133

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-158974

(43)Date of publication of application : 31.05.2002

---

(51)Int.Cl. H04N 5/93
G11B 20/10
G11B 20/12
H04N 5/85
H04N 5/92
H04N 7/24

---

(21)Application number : 2001-109341 (71)Applicant : SONY CORP

(22)Date of filing : 06.04.2001 (72)Inventor : KATO MOTOKI
HAMADA TOSHIYA

---

(30)Priority
Priority number : 2000183769
2000271550
Priority date : 21.04.2000
07.09.2000
Priority country : JP
JP

---

(54) INFORMATION PROCESSOR AND PROCESSING METHOD, AND RECORDING MEDIUM THEREFOR, AND PROGRAM AND ITS RECORDING MEDIUM

(57)Abstract:
PROBLEM TO BE SOLVED: To reproduce dynamic images recorded individually while sustaining continuity.
SOLUTION: When Clip1 and Clip2 recorded individually are reproduced continuously, a Bridge Clip playing a role of bridging from Clip1 to Clip2 is generated. The Bridge Clip comprises corresponding parts of the Clip1 and Clip2 where switching is made from Clip1 to Clip2.

CLAIMS

[Claim(s)]
[Claim 1]An information processor comprising:
When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, It comprises a predetermined portion of said 1st AV stream, and a predetermined portion of said 2nd AV stream, While generating the 3rd AV stream reproduced when reproduction is switched to said 2nd AV stream from said 1st AV stream, Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream.
A creating means which generates address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream.
Said 3rd AV stream generated by said creating means and a recording device which records said address information.

[Claim 2]An arrival time stamp of a source packet of said 1st AV stream contained in said address information generated by said creating means, An arrival time stamp of a source packet located in the beginning of said 3rd AV stream is continuing, And an arrival time stamp of a source packet of said 2nd AV stream contained in said address information generated by said creating means, The information processor according to claim 1, wherein an arrival time stamp of a source packet located in the last of said 3rd AV stream is continuing.
[Claim 3]The information processor according to claim 2, wherein only one break point exists in an arrival time stamp of a source packet in said 3rd AV stream.
[Claim 4]A data part of an AV stream before a source packet shown using information on an address of a source packet of said 1st AV stream contained in said address information generated by said creating means, The information processor according to claim 2 characterized by determining said address so that it may be arranged to a continuation field more than a predetermined size on a recording medium.
[Claim 5]A data part of an AV stream after a source packet shown using information on an address of a source packet of said 2nd AV stream contained in said address information generated by said creating means, The information processor according to claim 2 characterized by determining said address so that it may be arranged to a continuation field more than a predetermined size on a recording medium.
[Claim 6]The information processor according to claim 2 characterized by generating said 3rd AV stream so that said 3rd AV stream may be arranged to a continuation field

more than a predetermined size on a recording medium.
[Claim 7]An information processing method comprising:
When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, It comprises a predetermined portion of said 1st AV stream, and a predetermined portion of said 2nd AV stream, While generating the 3rd AV stream reproduced when reproduction is switched to said 2nd AV stream from said 1st AV stream, Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream.
A generation step which generates address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream.

[Claim 8]A recording medium with which a program which a computer can read is recorded, comprising:
When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, It comprises a predetermined portion of said 1st AV stream, and a predetermined portion of said 2nd AV stream, While generating the 3rd AV stream reproduced when reproduction is switched to said 2nd AV stream from said 1st AV stream, Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream.
A generation step which generates address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream.

[Claim 9]When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, It comprises a predetermined portion of said 1st AV stream, and a predetermined portion of said 2nd AV stream, While generating the 3rd AV stream reproduced when reproduction is switched to said 2nd AV stream from said 1st AV stream, Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream, A program which makes a computer perform a generation step which generates address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream.

[Claim 10]An information processor comprising:
The 1st reading means that reads the 1st AV stream, the 2nd AV stream, or the 3rd AV stream from a recording medium.
Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream.
The 2nd reading means that reads address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream from said recording medium.
Based on information relevant to said 3rd AV stream read by said 2nd reading means, A reproduction means which changes reproduction from said 1st AV stream read by said 1st reading means to said 3rd AV stream, changes reproduction from said 3rd AV stream to said 2nd AV stream, and is reproduced.

[Claim 11]An information processing method comprising:
The 1st reading control step that controls read-out from a recording medium of the 1st AV stream, the 2nd AV stream, or the 3rd AV stream.
Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information to which said 3rd AV stream relates.
The 2nd reading control step that controls read-out from said recording medium of address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream.
Based on information relevant to said 3rd AV stream by which read-out was controlled by processing of said 2nd reading control step, Regeneration steps which change reproduction from said 1st AV stream by which read-out was controlled by processing of said 1st reading control step to said 3rd AV stream, change reproduction from said 3rd AV stream to said 2nd AV stream, and are reproduced.

[Claim 12]A recording medium with which a program which a computer can read is recorded, comprising:
The 1st reading control step that controls read-out from a recording medium of the 1st AV stream, the 2nd AV stream, or the 3rd AV stream.
Information on an address of a source packet of said 1st AV stream in timing which

changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream.
The 2nd reading control step that controls read-out from said recording medium of address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream.
Based on information relevant to said 3rd AV stream by which read-out was controlled by processing of said 2nd reading control step, Regeneration steps which change reproduction from said 1st AV stream by which read-out was controlled by processing of said 1st reading control step to said 3rd AV stream, change reproduction from said 3rd AV stream to said 2nd AV stream, and are reproduced.

[Claim 13]The 1st AV stream, the 2nd AV stream, Or the 1st reading control step that controls read-out from a recording medium of the 3rd AV stream, Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream, The 2nd reading control step that controls read-out from said recording medium of address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream, Based on information relevant to said 3rd AV stream by which read-out was controlled by processing of said 2nd reading control step, Reproduction is changed from said 1st AV stream by which read-out was controlled by processing of said 1st reading control step to said 3rd AV stream, A program which makes a computer perform regeneration steps which change reproduction from said 3rd AV stream to said 2nd AV stream, and are reproduced.
[Claim 14]When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, The 3rd AV stream reproduced when it comprises a predetermined portion of said 1st AV stream, and a predetermined portion of said 2nd AV stream and reproduction is switched to said 2nd AV stream from said 1st AV stream, Information on an address of a source packet of said 1st AV stream in timing which changes reproduction from said 1st AV stream to said 3rd AV stream as information relevant to said 3rd AV stream, A recording medium, wherein address information which comprises information on an address of a source packet of said 2nd AV stream in timing which changes reproduction from said 3rd AV stream to said 2nd AV stream is recorded.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]this invention relates to an information processor and a method, a recording medium, a program, and a recording medium -- especially -- a reproducing section -- it is related with the information processor and the method, the recording medium, program, and recording medium which maintain the continuity of the video to kick.

[0002]

[Description of the Prior Art]In recent years, various kinds of optical discs are being proposed as a dismountable disk type recording medium from a recording and reproducing device. The optical disc in which such record is possible is proposed as several gigabytes of mass media.

The expectation as media which record AV (Audio Visual) signals, such as a video signal, is high.

As sauce (supply source) of the digital AV signal recorded on the optical disc in which this record is possible, there are CS digital satellite broadcasting and BS digital broadcasting, and the terrestrial television broadcasting of the digital system, etc. are

proposed in the future.

[0003]Here, as for the digital video signal supplied from these sauce, it is common that graphical data compression is usually carried out by MPEG(Moving Picture Experts Group) 2 method. The recording rate peculiar to the device is provided in the recorder. By the conventional noncommercial image storage medium, if it is an analog recording method when recording the digital video signal of digital broadcasting origin, after decoding a digital video signal, a band limit will be carried out and it will record. Or if it is digital recording systems including MPEG1 Video, MPEG 2 Video, and DV method, after being decoded once, with a recording rate and a coding mode peculiar to the device, it will be re-encoded and will be recorded.

[0004]However, such a record method decodes the supplied bit stream once, and it is accompanied by degradation of image quality in order to record by performing band limit and re-encoding after that. When the transmission rate of the digital signal inputted when the digital signal by which graphical data compression was carried out was recorded does not exceed the recording rate of a recording and reproducing device, decoding and the method of recording as it is, without re-encoding have least degradation of image quality in the supplied bit stream. However, when the transmission rate of the digital signal by which graphical data compression was carried out exceeds the recording rate of the disk as a recording medium, it is necessary to carry out re-encoding and to record so that a transmission rate may become below a maximum of the recording rate of a disk after decoding with a recording and reproducing device.

[0005]When the bit rate of the input digital signal is transmitted by the variable rate method fluctuated by time, Since a rotary head is fixed number of rotations, a recording rate stores data in a buffer once compared with the tape recording system which becomes a fixed rate, and the disc recording device for which record can be done burstily can use the capacity of a recording medium without futility.

[0006]As mentioned above, in the future which becomes in use, digital broadcasting is predicted that decoding and the recording and reproducing device which recorded without re-encoding and uses a disk as recording media are asked for a broadcasting signal like a data streamer with a digital signal.

[0007]

[Problem(s) to be Solved by the Invention]When reproducing the data recorded on the recording medium in a recorder which was mentioned above, the ******** skip reproduction of reproducing to a predetermined picture, continuing the picture located in the position which is separated from the picture in time, and reproducing occurs.

the continuity time when skip reproduction is performed on the image to reproduce breaks off -- stripes -- the technical problem that there were things occurred.
[0008]this invention is made in view of such a situation -- a reproducing section -- it aims at enabling it to reproduce so that the continuity of the video to kick may be maintained.
[0009]
[Means for Solving the Problem]This invention is characterized by the 1st information processor comprising the following.
When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, While generating the 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream, and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream.
A creating means which generates address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream.
The 3rd AV stream generated by creating means and a recording device which records address information.

[0010]An arrival time stamp of a source packet of the 1st AV stream contained in address information generated by said creating means, An arrival time stamp of a source packet located in the beginning of the 3rd AV stream is continuing, And an arrival time stamp of a source packet of the 2nd AV stream contained in address information generated by creating means and the arrival time stamp of a source packet located in the last of the 3rd AV stream can be continuing.
[0011]Only one break point can exist in an arrival time stamp of a source packet in said 3rd AV stream.
[0012]A data part of an AV stream before a source packet shown using information on an address of a source packet of the 1st AV stream contained in address information generated by said creating means, An address can be determined so that it may be arranged to a continuation field more than a predetermined size on a recording medium.
[0013]A data part of an AV stream after a source packet shown using information on an address of a source packet of the 2nd AV stream contained in address information

generated by said creating means, An address can be determined so that it may be arranged to a continuation field more than a predetermined size on a recording medium.

[0014]The 3rd AV stream can be generated so that said 3rd AV stream may be arranged to a continuation field more than a predetermined size on a recording medium.

[0015]When it is directed that the 1st information processing method of this invention is continuously reproduced from the 1st AV stream to the 2nd AV stream, While generating the 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream, and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, A generation step which generates address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream is included.

[0016]When it is directed that a program of the 1st recording medium of this invention is continuously reproduced from the 1st AV stream to an AV stream of **, While generating the 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream, and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, A generation step which generates address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream is included.

[0017]When it is directed that the 1st program of this invention is continuously reproduced from the 1st AV stream to an AV stream of **, While generating the 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream, and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, A computer is made to perform a generation step which generates address information

which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream.
[0018]This invention is characterized by the 2nd information processor comprising the following.
The 1st reading means that reads the 1st AV stream, the 2nd AV stream, or the 3rd AV stream from a recording medium.
Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream.
The 2nd reading means that reads address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream from a recording medium.
Based on information relevant to the 3rd AV stream read by the 2nd reading means, A reproduction means which changes reproduction from the 1st AV stream read by the 1st reading means to the 3rd AV stream, changes reproduction from the 3rd AV stream to the 2nd AV stream, and is reproduced.

[0019]This invention is characterized by the 2nd information processing method comprising the following.
The 1st reading control step that controls read-out from a recording medium of the 1st AV stream, the 2nd AV stream, or the 3rd AV stream.
Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream.
The 2nd reading control step that controls read-out from a recording medium of address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream.
Based on information relevant to the 3rd AV stream by which read-out was controlled by processing of the 2nd reading control step, Regeneration steps which change reproduction from the 1st AV stream by which read-out was controlled by processing of the 1st reading control step to the 3rd AV stream, change reproduction from the 3rd AV stream to the 2nd AV stream, and are reproduced.

[0020]This invention is characterized by a program of the 2nd recording medium comprising the following.

The 1st reading control step that controls read-out from a recording medium of the 1st AV stream, the 2nd AV stream, or the 3rd AV stream.
Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream.
The 2nd reading control step that controls read-out from a recording medium of address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream.
Based on information relevant to the 3rd AV stream by which read-out was controlled by processing of the 2nd reading control step, Regeneration steps which change reproduction from the 1st AV stream by which read-out was controlled by processing of the 1st reading control step to the 3rd AV stream, change reproduction from the 3rd AV stream to the 2nd AV stream, and are reproduced.

[0021]The 2nd program of this invention The 1st AV stream, the 2nd AV stream, Or the 1st reading control step that controls read-out from a recording medium of the 3rd AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, The 2nd reading control step that controls read-out from a recording medium of address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream, Based on information relevant to the 3rd AV stream by which read-out was controlled by processing of the 2nd reading control step, A computer is made to perform regeneration steps which change reproduction from the 1st AV stream by which read-out was controlled by processing of the 1st reading control step to the 3rd AV stream, change reproduction from the 3rd AV stream to the 2nd AV stream, and are reproduced.
[0022]When it is directed that the 3rd recording medium of this invention is continuously reproduced from the 1st AV stream to the 2nd AV stream, The 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream, and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, Address

information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream is recorded.

[0023]In the 1st information processor of this invention, a method, and a program, When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, While the 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream is generated, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, Address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream is generated.

[0024]In the 2nd information processor of this invention, a method, and a program, the 1st AV stream, the 2nd AV stream, Or the 3rd AV stream is read from a recording medium, and as information relevant to the 3rd AV stream, Information on an address of a source packet of the 1st AV stream in timing which changes reproduction from the 1st AV stream to the 3rd AV stream, Address information which comprises information on an address of a source packet of the 2nd AV stream in timing which changes reproduction from the 3rd AV stream to the 2nd AV stream is read from a recording medium, Based on information relevant to the 3rd read AV stream, reproduction is changed from the 1st AV stream to the 3rd AV stream, reproduction is changed from the 3rd AV stream to the 2nd AV stream, and it is reproduced.

[0025]

[Embodiment of the Invention]Below, an embodiment of the invention is described with reference to drawings. Drawing 1 is a figure showing the example of an internal configuration of the recording and reproducing device 1 which applied this invention. First, the composition of the portion which performs operation which records the signal inputted from the outside on a recording medium is explained. The recording and reproducing device 1 is considered as the composition which can input analog data or digital data and can be recorded.

[0026]The video signal of an analog is inputted into the terminal 11, and the audio signal of an analog is inputted into the terminal 12, respectively. The video signal inputted into the terminal 11 is outputted to the analyzing parts 14 and the AV encoder 15, respectively. The audio signal inputted into the terminal 12 is outputted to the AV encoder 15. The analyzing parts 14 extract the focus, such as a scene change,

from the inputted video signal.

[0027]The AV encoder 15 codes the video signal and audio signal which were inputted, respectively, and outputs system information (S), such as coding video stream (V), a coding audio stream (A), and AV synchronization, to the multiplexer 16.

[0028]A coding video stream is a video stream coded by MPEG(Moving Picture Expert Group) 2 method, for example, Coding audio streams are the audio stream coded by MPEG1 method, an audio stream coded by Dolbey AC3 method, etc., for example. The multiplexer 16 multiplexes the stream of the inputted video and an audio based on input system information, and outputs it to the multiplexed stream analyzing parts 18 and saw spa KETTAIZA 19 via the switch 17.

[0029]Multiplexed streams are an MPEG2 transport stream and an MPEG 2 program stream, for example. Saw spa KETTAIZA 19 codes the AV stream which comprises a source packet in the inputted multiplexed stream according to the application format of the recording medium 100 on which the stream is made to record. Predetermined processing is performed by the ECC (error correction) coding part 20 and the modulation part 21, and an AV stream is outputted to the writing part 22. The writing part 22 writes an AV stream file in the recording medium 100 based on the control signal outputted from the control section 23 (it records).

[0030]Transport streams, such as digital television broadcasting inputted from a digital interface or a digital television tuner, are inputted into the terminal 13. They are a method which records those with two kind, and them on a transparent at the recording method of a transport stream inputted into the terminal 13, and a method recorded after carrying out re-encoding for the purposes, such as lowering a recording bit rate. The directions information on a recording method is inputted into the control section 23 from the terminal 24 as a user interface.

[0031]When recording an input transport stream on a transparent, the transport stream inputted into the terminal 13 is outputted to the multiplexed stream analyzing parts 18 and saw spa KETTAIZA 19. Since processing until an AV stream is recorded on the recording medium 100 after this is the same processing as the case where above-mentioned input audio osmosis and a video signal are coded and recorded, the explanation is omitted.

[0032]When recording after re-encoding an input transport stream, the transport stream inputted into the terminal 13 is inputted into the demultiplexer 26. The demultiplexer 26 performs demultiplex processing to the inputted transport stream, and extracts video stream (V), an audio stream (A), and system information (S).

[0033]A video stream is outputted to AV decoder 27 among the streams (information)

extracted by the demultiplexer 26, and an audio stream and system information are outputted to the multiplexer 16, respectively. AV decoder 27 decodes the inputted video stream, and outputs the reproduced video signal to the AV encoder 15. The AV encoder 15 codes an input video signal, and outputs coding video stream (V) to the multiplexer 16.

[0034]The audio stream which was outputted from the demultiplexer 26 and inputted into the multiplexer 16 on the other hand, and system information, And the video stream outputted from the AV encoder 15 is multiplexed based on input system information, and is outputted to the multiplexed stream analyzing parts 18 and source packet TAIZA 19 via the switch 17 as a multiplexed stream. Since processing until an AV stream is recorded on the recording medium 100 after this is the same processing as the case where an above-mentioned input audio signal and video signal are coded and recorded, the explanation is omitted.

[0035]The recording and reproducing device 1 of this embodiment records the file of an AV stream on the recording medium 100, and it also records the application data base information explaining the file. Application data base information is created by the control section 23. The input to the control section 23 is the characteristic information of the video from the analyzing parts 14, the characteristic information of the AV stream from the multiplexed stream analyzing parts 18, and directions information from a user that it is inputted from the terminal 24.

[0036]The characteristic information of the video supplied from the analyzing parts 14, It is the information related to the characteristic picture in an input dynamic image signal, for example, is specification information (mark), including the starting point of a program, a scene change point, a start, an end point of commercials (CM), etc., and the information on the thumbnail image of the picture of the designation location is also included.

[0037]The characteristic information of the AV stream from the multiplexed stream analyzing parts 18, It is the information related to the encoded information of the AV stream recorded. For example, they are the change point information of the address information of I picture in an AV stream, the encoding parameter of an AV stream, and the encoding parameter in an AV stream, the information (mark) related to the characteristic picture in a video stream, etc.

[0038]The directions information of the user from the terminal 24 is a bookmark, information on resume points, etc. which the character character and user explaining the specification information on the reproducing section specified by the user in an AV stream and the contents of the reproducing section set to a favorite scene.

[0039]Based on the above-mentioned input, the control section 23 The database of an AV stream (Clip), The management information (info.dvr) of the database of what (PlayList) carried out grouping of the reproducing section (PlayItem) of an AV stream, and the contents of record of the recording medium 100, and the information on a thumbnail image are created. Like an AV stream, the application data base information which comprises these information is processed by the ECC code-ized part 20 and the modulation part 21, and is inputted into the writing part 22. The writing part 22 records a database file on the recording medium 100 based on the control signal outputted from the control section 23.

[0040]The details about the application data base information mentioned above are mentioned later.

[0041]Thus, the AV stream file (file of image data and voice data) recorded on the recording medium 100, When application data base information is reproduced, the control section 23 directs to read application data base information from the recording medium 100 to the read section 28 first. And the read section 28 reads application data base information from the recording medium 100, and the application data base information is inputted into the control section 23 through processing of the demodulation section 29 and the ECC decoding part 30.

[0042]The control section 23 outputs the list of PlayList currently recorded on the recording medium 100 to the user interface of the terminal 24 based on application data base information. A user chooses PlayList to reproduce from the list of PlayList, and the information about PlayList which had reproduction specified is inputted into the control section 23. The control section 23 directs read-out of an AV stream file required for reproduction of the PlayList to the read section 28. The read section 28 reads an AV stream corresponding from the recording medium 100 according to the directions, and outputs it to the demodulation section 29. It gets over by performing predetermined processing, and the AV stream inputted into the demodulation section 29 is further outputted sauce DEPAKETTAIZA 31 through processing of the ECC decoding part 30.

[0043]Sauce DEPAKETTAIZA 31 is read from the recording medium 100, and is changed into the stream which can output the AV stream of an application format to which predetermined processing was performed to the demultiplexer 26. The demultiplexer 26 outputs system information (S), such as video stream (V) which constitutes the reproducing section (PlayItem) of an AV stream specified by the control section 23, an audio stream (A), and AV synchronization, to AV decoder 27. AV decoder 27 decodes a video stream and an audio stream, and outputs a

reproduced video signal and a reproduced audio signal from the terminal 32 corresponding, respectively and the terminal 33.

[0044]When the information which directs random access reproduction and special reproduction is inputted from the terminal 24 as a user interface, the control section 23, Based on the contents of the database (Clip) of an AV stream, the reading position of the AV stream from the storage 100 is determined, and read-out of the AV stream is directed to the read section 28. For example, when reproducing PlayList with the selected user from predetermined time, the control section 23 directs to read the data with the time stamp nearest to the specified time from I picture to the read section 28.

[0045]When fast reproduction (Fast-forward playback) is directed by the user, the control section 23, Based on the database (Clip) of an AV stream, it directs to read I-picture data in an AV stream continuously one by one to the read section 28.

[0046]The read section 28 reads the data of an AV stream from the specified random access point, and the read data is reproduced through processing of latter each part.

[0047]Next, a user explains the case where the AV stream currently recorded on the recording medium 100 is edited. When a user wants to specify the reproducing section of the AV stream currently recorded on the recording medium 100, and to create new salvage pathway, For example, from the popular music show of the program A, reproduce the singer's A portion and it continues after that, The information on the starting point (yne point) of a reproducing section and an end point (out point) is inputted into the control section 23 from the terminal 24 as a user interface to create the salvage pathway of liking to reproduce the portion of the singer A of the popular music show of the program B. The control section 23 creates the database of what (PlayList) carried out grouping of the reproducing section (PlayItem) of an AV stream.

[0048]When a user wants to eliminate a part of AV stream currently recorded on the recording medium 100, the information on the yne point of the elimination section and an out point is inputted into the control section 23 from the terminal 24 as a user interface. The control section 23 changes the database of PlayList so that only a required AV stream portion may be referred to. It directs to the writing part 22 so that the unnecessary stream portion of an AV stream may be eliminated.

[0049]It is a case where a user wants to specify the reproducing section of the AV stream currently recorded on the recording medium 100, and to create new salvage pathway, and the case where he would like to connect each reproducing section seamlessly is explained. In such a case, the control section 23 creates the database of what (PlayList) carried out grouping of the reproducing section (PlayItem) of an AV

stream, and performs partial re-encoding and re-multiplex-izing of a reproducing section of the video stream near a node further.

[0050]First, the information on the picture of the yne point of a reproducing section and the information on the picture of an out point are inputted into the control section 23 from the terminal 24. The control section 23 directs read-out of data required in order to reproduce the yne point side picture and the picture by the side of an out point to the read section 28. And the read section 28 reads data from the recording medium 100, and the data is outputted to the demultiplexer 26 through the demodulation section 29, the ECC decoding part 30, and sauce DEPAKETTAIZA 31.

[0051]The control section 23 analyzes the data inputted into the demultiplexer 26, A re multiplex-ized method is determined as the re-encoding method (change of picture_coding_type, assignment of the re-encoded encoding bit amount) of a video stream, and the method is supplied to the AV encoder 15 and the multiplexer 16.

[0052]Next, the demultiplexer 26 divides the inputted stream into video stream (V), an audio stream (A), and system information (S). A video stream has "data inputted into AV decoder 27", and "the data inputted into the multiplexer 16." It is data required in order to re-encode the former data, and this is decoded by AV decoder 27, and the decoded picture is re-encoded with the AV encoder 15, and is made into a video stream. The latter data is data copied from an original stream without carrying out re-encoding. About an audio stream and system information, it is directly inputted into the multiplexer 16.

[0053]Based on the information inputted from the control section 23, the multiplexer 16 multiplexes an input stream and outputs a multiplexed stream. A multiplexed stream is processed by the ECC code-ized part 20 and the modulation part 21, and is inputted into the writing part 22. The writing part 22 records an AV stream on the recording medium 100 based on the control signal supplied from the control section 23.

[0054]Explanation about operation of the reproduction and edit based on application data base information and its information is given to below. Drawing 2 is a figure explaining the structure of an application format. An application format has two layers, PlayList and Clip, for management of an AV stream. Volume Information carries out management of all the Clip(s) and PlayList(s) in a disk. Here, the pair of one AV stream and its attached information is considered to be one object, and it is called Clip. An AV stream file calls Clip AV stream file, and the attached information is called Clip Informationfile.

[0055]One Clip AV stream file stores the data which has arranged the MPEG2 transport stream in the structure in which it is specified by application format.

Generally, although a file is treated as a sequence of bytes, the contents of Clip AV stream file are developed on a time-axis, and the entry point in Clip is mainly specified in a hourly base. When the time stamp of the access point to predetermined Clip is given, Clip Information file is useful in order to find the address information which should start read-out of data in Clip AV stream file.

[0056]PlayList is explained with reference to drawing 3. PlayList chooses from Clip(s) the reproducing section which a user wants to see, and it is provided in order to be able to edit it easily. One PlayList is a meeting of the reproducing section in Clip. One reproducing section in predetermined Clip is called PlayItem, and it is expressed with the pair of the yne point (IN) on a time-axis, and an out point (OUT). Therefore, PlayList is constituted when two or more PlayItem(s) gather.

[0057]There are two types of PlayList(s). One is Real PlayList and another is Virtual PlayList. Real PlayList is sharing the stream portion of Clip which it is referring to. That is, when Real PlayList occupies in a disk the data volume equivalent to the stream portion of Clip which is referring to it and Real PlayList is eliminated, data is eliminated also for the stream portion of Clip which it is referring to.

[0058]Virtual PlayList is not sharing the data of Clip. Therefore, even if Virtual PlayList is changed or eliminated, by the contents of Clip, change does not arise at all.

[0059]Next, edit of Real PlayList is explained. Drawing 4 (A) is a figure about the creation (create: creation) of Real PlayList, and when an AV stream is recorded as new Clip, it is operation in which RealPlayList which refers to the whole Clip is newly created.

[0060]Drawing 4 (B) is a figure about the divide (divide: division) of Real PlayList, and is operation in which Real PlayList is divided at a point [****] and divided into two Real PlayList. For example in one clip managed by one PlayList, when two programs are managed, in registration (record), a user does the operation of this division again as each program, and it is performed at the time of being **********. There is nothing for which the contents of Clip are changed by this operation (the Clip itself is divided).

[0061]Drawing 4 (C) is a figure about the combine (combine: combination) of Real PlayList, and is operation which combines two Real PlayList and is set to one new Real PlayList. A user reregisters two programs as one program, and operation of this combination is performed at the time of being **********, for example. There is nothing for which Clip is changed by this operation (the Clip itself is set to one).

[0062]Drawing 5 (A) is a figure about deletion (delete: deletion) of whole Real PlayList,

When operation which eliminates whole predetermined Real PlayList is carried out, the stream portion to which Clip which deleted Real PlayList refers to corresponds is also deleted.

[0063]Drawing 5 (B) is a figure about partial deletion of Real PlayList, and when a portion [**** / Real PlayList] is deleted, it is changed so that corresponding PlayItem may refer to only the stream portion of required Clip. And the stream portion to which Clip corresponds is deleted.

[0064]Drawing 5 (C) is a figure about minimization (Minimize: minimization) of Real PlayList, It is operation of referring to only the stream portion of Clip required for Virtual PlayList for PlayItem corresponding to Real PlayList. The stream portion to which Clip unnecessary for Virtual PlayList corresponds is deleted.

[0065]Real PlayList is changed by the operation which was mentioned above, When the stream portion of Clip which the Real PlayList refers to is deleted, Virtual PlayList which is using the deleted Clip may exist, and a problem may arise by deleted Clip in the Virtual PlayList.

[0066]As opposed to operation of [so that such a thing may not arise] deletion to a user, "If Virtual PlayList which is referring to the stream portion of Clip which the Real PlayList is referring to exists and the Real PlayList is eliminated, although the Virtual PlayList will also be eliminated, is it still good? processing of the deletion with a user's directions after urging a check (warning) by displaying the message " etc. -- execution -- or it cancels. Or operation of minimization is made to be performed instead of deleting Virtual PlayList to Real PlayList.

[0067]Next, the operation to Virtual PlayList is explained. The contents of Clip are not changed even if operation is performed to Virtual PlayList. Drawing 6 is assembling (Assemble). Edit (IN-OUT edit) It is a related figure and is operation of making PlayItem of the reproducing section for which it asked when the user wanted to see, and creating Virtual PlayList. The seamless connection between PlayItem(s) is supported by the application format (after-mentioned).

[0068]As shown in drawing 6 (A), two Real PlayList1 and 2, When Clip1 corresponding to each RealPlayList and 2 exist, A user points to the predetermined section (section-layItem1 to In1 thru/or Out1) in Real PlayList1 as a reproducing section, and as the section reproduced continuously, When it points to the predetermined section (section-layItem2 to In2 thru/or Out2) in Real PlayList2 as a reproducing section, As shown in drawing 6 (B), one Virtual PlayList which comprises PlayItem1 and PlayItem2 is created.

[0069]Next, the reorganization collection (Re-editing) of Virtual PlayList is explained.

In a reorganization collection, change of the yne point in Virtual PlayList, and an out point, There are insertion (insert) of new PlayItem to Virtual PlayList, an addition (append), deletion of PlayItem in Virtual PlayList, etc. Virtual PlayList itself can also be deleted.

[0070]Drawing 7 is a figure about postrecording (Audio dubbing (post recording)) of the audio to Virtual PlayList, and is operation which registers postrecording of the audio to VirtualPlayList as a sub path. Postrecording of this audio is supported by the application format. An additional audio stream is added to the AV stream of the main path of Virtual PlayList as a sub path.

[0071]As operation common to Real PlayList and Virtual PlayList, there is change (Moving) of the reproduction sequence of PlayList as shown in drawing 8. This operation is change of the reproduction sequence of PlayList in the inside of a disk (volume), and is supported by Table Of PlayList (with reference to drawing 20 etc., it mentions later) defined in an application format. The contents of Clip are not changed by this operation.

[0072]Next, the mark (Mark) is explained. The mark is provided in order to specify the highlight in Clip and PlayList, and characteristic time. Specify the characteristic scene resulting from the contents of the AV stream, for example, the mark added to Clip is a scene change point etc. When reproducing PlayList, it can be used with reference to the mark of Clip which the PlayList refers to.

[0073]Are mainly set by the user, for example, the marks added to PlayList are a bookmark, resume points, etc. Setting a mark to Clip or PlayList is performed by adding the time stamp in which the time of a mark is shown to a mark list. Deleting a mark is removing the time stamp of the mark out of a mark list. Therefore, as for an AV stream, a change of what is not made by setting out or deletion of a mark, either.

[0074]Next, a thumbnail is explained. A thumbnail is a still picture added to Volume, PlayList, and Clip. There are two kinds of thumbnails and one is a thumbnail as representation drawing showing the contents. This is used by the menu screen for choosing the thing a user mainly wants to operate and look at cursor (un-illustrating) etc. Another is a picture showing the scene which the mark has pointed out.

[0075]Volume and each Playlist need to enable it to have representation drawing. The representation drawing of Volume is a disk (the recording medium 100 presupposes that it is a disk-like thing, and the recording medium 100 and the following). suitably -- a disk -- describing -- when it sets to the predetermined place of the recording and reproducing device 1, it assumes being used when displaying the still picture showing the contents of the disk first. In the menu screen which chooses Playlist, the

representation drawing of Playlist assumes being used as a still picture for expressing the contents of Playlist.

[0076]Although it is possible as representation drawing of Playlist to make the picture of the beginning of Playlist into a thumbnail (representation drawing), when the picture of the head of the regeneration time 0 expresses the contents, it is not not necessarily the optimal picture. Then, a user enables it to set up arbitrary pictures as a thumbnail of Playlist. Two kinds of thumbnails are called a menu thumbnail above. Since a menu thumbnail is displayed frequently, it needs to be read from a disk at high speed. For this reason, it is efficient to store all the menu thumbnails in one file. It is not necessary to be necessarily the picture extracted from the animation in volume, and as shown in drawing 10, a menu thumbnail may be taken from a personal computer or a digital still camera, and a ********** picture may be sufficient as it.

[0077]It can be necessary to strike two or more marks, and in order to know the contents of the mark position, it is necessary to enable it to see the picture of a marking point easily to Clip and Playlist on the other hand. The picture showing such a marking point is called the mark thumbnail (Mark Thumbnails). Therefore, the picture which becomes the origin of a thumbnail becomes more nearly main [what extracted the picture of the marking point] than the picture captured from the exterior.

[0078]Drawing 11 is a mark attached to PlayList, and a figure showing the relation of the mark thumbnail, and drawing 12 is a mark attached to Clip, and a figure showing the relation of the mark thumbnail. Since a mark thumbnail is used with a sub menu etc. when the details of Playlist are expressed unlike a menu thumbnail, what it is read in short access time is not required. Therefore, whenever a thumbnail is needed, the recording and reproducing device 1 opens a file, and it does not become a problem even if it takes time somewhat by reading a part of the file.

[0079]In order to reduce the number of files which exists in volume, all the mark thumbnails are good to store in one file. Although Playlist can have one menu thumbnail and two or more mark thumbnails, since Clip does not have the necessity that a direct user chooses (it usually specifies via Playlist), it does not need to provide a menu thumbnail.

[0080]Drawing 13 is a figure showing the relation of the menu thumbnail at the time of taking having mentioned above into consideration, a mark thumbnail, PlayList, and Clip. The menu thumbnail provided in the menu thumbnail file for every PlayList is filed. The volume thumbnail representing the contents of the data currently recorded on the disk is contained in the menu thumbnail file. The thumbnail by which the mark thumbnail file was created for every PlayList and every Clip is filed.

[0081]Next, CPI (Characteristic Point Information) is explained. CPI is data contained in Clip information files, When the time stamp of the access point to Clip is given, it is mainly used in order to find the data address which should start read-out of data in Clip AV stream file. According to this embodiment, two kinds of CPI(s) are used. One is EP_map and another is TU_map.

[0082]EP_map is a list of entry point (EP) data, and it is extracted from an elementary stream and a transport stream. This has the address information for finding the place of the entry point which should start decoding in an AV stream. One EP data comprises a pair of the data address in the AV stream of a presentation time stamp (PTS) and the access unit corresponding to the PTS.

[0083]EP_map is mainly used for two purposes. It is used in order to find the data address in the AV stream of the access unit referred to [1st] with a presentation time stamp in PlayList. It is used for the 2nd for first forward reproduction or first reverse reproduction. When the recording and reproducing device 1 records an input AV stream and the syntax of the stream can be analyzed, EP_map is created and it is recorded on a disk.

[0084]TU_map has a list of the time unit (TU) data based on the arrival time of the transport packet inputted through a digital interface. This gives the relation between the time of an arrival time base, and the data address in an AV stream. When the recording and reproducing device 1 records an input AV stream and the syntax of the stream cannot be analyzed, TU_map is created and it is recorded on a disk.

[0085]This embodiment defines the stream format (SESF) of self encoding. SESF is used when coding to an MPEG2 transport stream, after decoding the purpose of coding an analog input signal, and a digital input signal (for example, DV).

[0086]SESF defines coding restrictions of the elementary stream about MPEG-2 transport stream and an AV stream. When the recording and reproducing device 1 encodes and records a SESF stream, EP_map is created and it is recorded on a disk.

[0087]Either of the methods shown below is used and the stream of digital broadcasting is recorded on the recording medium 100. First, transformer coding of the stream of digital broadcasting is carried out at a SESF stream. In this case, the recorded stream must be based on SESF. In this case, EP_map must be created and it must be recorded on a disk.

[0088]Or transformer coding is carried out at new elementalist ream, and the elementary stream which constitutes a digital broadcasting stream is re-multiplex-ized to the new transport stream based on the stream format which the standardization organization of the digital broadcasting stream defines. In this case,

EP_map must be created and it must be recorded on a disk.

[0089]For example, an input stream is MPEG-2 transport stream of ISDB (standard name of digital BS broadcasting of Japan) conformity, and suppose that it contains a HDTV video stream and a MPEG AAC audio stream. Transformer coding of the HDTV video stream is carried out at a SDTV video stream, and the SDTV video stream and an original AAC audio stream are re-multiplex-ized to TS. Both the transport streams recorded as a SDTV stream must be based on an ISDB format.

[0090]The stream of digital broadcasting as other methods at the time of being recorded on the recording medium 100, It is a case (it records without changing any input transport streams) where an input transport stream is recorded on a transparent, and EP_map is then created and it is recorded on a disk.

[0091]Or it is a case (it records without changing any input transport streams) where an input transport stream is recorded on a transparent, and TU_map is then created and it is recorded on a disk.

[0092]Next, a directory and a file are explained. Hereafter, the recording and reproducing device 1 is suitably described to be DVR (Digital Video Recording). Drawing 14 is a figure showing an example of the directory structure on a disk. A directory required on the disk of DVR, As shown in drawing 14, they are a root directory including a "DVR" directory, a "PLAYLIST" directory, a "CLIPINF" directory, and a "DVR" directory including an "M2TS" directory and "DATA" directory. Although directories other than these may be made to be created under a root directory, they presuppose that it is ignored in the application format of this embodiment.

[0093]All the files and directories which are specified by DVR application format under a "DVR" directory are stored. A "DVR" directory includes four directories. Under a "PLAYLIST" directory, the database file of Real PlayList and Virtual PlayList is placed. This directory exists, even if one does not have PlayList.

[0094]The database of Clip is placed under a "CLIPINF" directory. This directory also exists, even if one does not have Clip. An AV stream file is placed under an "M2TS" directory. This directory exists, even if one does not have an AV stream file. In the "DATA" directory, the file of data broadcasting, such as digital TV broadcasting, is stored.

[0095]A "DVR" directory stores the file shown below. It is made under a "info.dvr" file and a DVR directory, and the overall information on an application layer is stored. Only one info.dvr must be under a DVR directory. A file name presupposes that it is fixed to info.dvr. A "menu.thmb" file stores the information relevant to a menu thumbnail

image. Zero or one menu thumbnail must be under a DVR directory. A file name presupposes that it is fixed to memu.thmb. When one does not have a menu thumbnail image, this file does not need to exist.

[0096]A "mark.thmb" file stores the information relevant to a mark thumbnail image. Zero or one mark thumbnail must be under a DVR directory. A file name presupposes that it is fixed to mark.thmb. When one does not have a menu thumbnail image, this file does not need to exist.

[0097]A "PLAYLIST" directory stores two kinds of PlayList files, and they are Real PlayList and VirtualPlayList. "xxxxx.rpls" A file stores the information relevant to one Real PlayList. One file is made for every Real PlayList. A file name is "xxxxx.rpls." Here, "xxxxx" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "rpls".

[0098]A "yyyyy.vpls" file stores the information relevant to one Virtual PlayList. One file is made for every Virtual PlayList. A file name is "yyyyy.vpls." Here, "yyyyy" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "vpls".

[0099]A "CLIPINF" directory stores one file corresponding to each AV stream file. "zzzzz.clpi" A file is Clip Information file corresponding to one AV stream file (Clip AV stream file or Bridge-Clip AV stream file). A file name is "zzzzz.clpi" and "zzzzz" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "clpi".

[0100]An "M2TS" directory stores the file of an AV stream. A "zzzzz.m2ts" file is an AV stream file treated by a DVR system. This is Clip AV stream file or Bridge-Clip AV stream. A file name is "zzzzz.m2ts" and "zzzzz" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "m2ts."

[0101]STCInfo stores the discontinuous dot data of STC in the AV stream file which is storing the MPEG2 transport stream. Temporarily, when an AV stream has a break point of STC, PTS of the same value may appear in the AV stream file. Therefore, when pointing out a certain time on an AV stream on a PTS basis, just PTS of an access point is insufficient in order to specify the point.

[0102]The index of the STC section [****] containing the PTS is required. The STC section [****] is called STC-sequence in this format, and that index is called STC-sequence-id. The information on STC-sequence is defined by STCInfo of Clip Information file. STC-sequence-id is an option in the AV stream file which uses it by an AV stream file with EP_map, and has TU_map.

[0103]A program is a meeting of elementalist ream and shares only one system time base for the synchronous reproduction of these streams. What the contents of the AV stream understand in advance of decoding of an AV stream for the recording and

reproducing device 1 is useful. For example, they are information, including the value of PID of the transport packet which transmits the elementary stream of video or an audio, video, the component kind of audio, etc., (for example, the videos of HDTV, the audio streams of MPEG-2 AAC, etc.).

[0104]This information is useful although the menu screen which explains to a user the contents of PlayList which refers to an AV stream is created, and, In advance of decoding of an AV stream, it is useful in order to set the AV decoder of playback equipment, and the initial state of a demultiplexer. For this reason, Clip Information file has ProgramInfo for explaining the contents of the program.

[0105]In the AV stream file which is storing the MPEG2 transport stream, the contents of a program may change in a file. For example, it is that PID of the transport packet which transmits a video elementary stream changes, or the component kind of video stream changes from SDTV to HDTV etc.

[0106]ProgramInfo stores the information on the change point of the contents of a program in the inside of an AV stream file. In an AV stream file, the contents of a program defined in this format call the fixed section Program-sequence. Program-sequence is an option in the AV stream file which uses it by an AV stream file with EP_map, and has TU_map.

[0107]The "DATA" directory stores the data transmitted from data broadcasting, and data is XML file, an MHEG file, etc., for example.

[0108]Next, the syntax and semantics of each directory (file) are explained. First, a "info.dvr" file is explained. Drawing 15 is a figure showing the syntax of a "info.dvr" file. A "info.dvr" file comprises three objects and they are DVRVolume(), TableOfPlayLists(), and MakerPrivateData().

[0109]To explain the syntax of info.dvr shown in drawing 15 TableOfPlayLists_Start_address, The start address of TableOfPlayList() is shown by making the relative number of bytes from the byte of the head of an info.dvr file into a unit. A relative number of bytes is counted from zero.

[0110]MakerPrivateData_Start_address shows the start address of MakerPrivateData() by making the relative number of bytes from the byte of the head of an info.dvr file into a unit. A relative number of bytes is counted from zero. padding_word (padding word) is inserted according to the syntax of info.dvr. N1 and N2 are zero or arbitrary positive integers. It may be made for each padding word to take any value.

[0111]DVRVolume() stores the information which describes the contents of volume (disk). Drawing 16 is a figure showing the syntax of DVRVolume(). version_number

shows four character characters which show the version number of this DVRVolume() for explaining the syntax of DVR Volume() shown in drawing 16. version_number is coded with "0045" according to ISO 646.

[0112]length is expressed with the 32-bit unsigned integer which shows the number of bytes of DVRVolume() from immediately after this length field to the last of DVRVolume().

[0113]ResumeVolume() has memorized the file name of Real PlayList reproduced at the end in volume, or Virtual PlayList. However, when a user interrupts reproduction of Real PlayList or Virtual PlayList, a playback position is stored in resume-mark defined in PlayListMark().

[0114]Drawing 17 is a figure showing the syntax of ResumeVolume(). The syntax of ResumeVolume() shown in drawing 17 to explain valid_flag, When it is shown that the resume_PlayList_name field is effective when this 1-bit flag is set to 1 and this flag is set to 0, it is shown that the resume_PlayList_name field is invalid.

[0115]10 bytes of field of resume_PlayList_name shows the file name of Real PlayList by which resume should be carried out, or Virtual PlayList.

[0116]UIAppInfoVolume in the syntax of DVRVolume() shown in drawing 16 stores the parameter of the user interface application about volume. Drawing 18 is a figure showing the syntax of UIAppInfoVolume, and the 8-bit field of character_set shows the encoding method of the character character coded in the Volume_name field for explaining the semantics. The encoding method corresponds to the value shown in drawing 19.

[0117]Eight bit fields of name_length show the byte length of the volume name shown in the Volume_name field. The field of Volume_name shows the name of volume. The number of bytes of the left in this field to a name_length number is an effective character character, and it shows the name of volume. In the Volume_name field, what kind of value may be [value after these effective character character] contained.

[0118]Volume_protect_flag is a flag which shows whether the contents in volume may be shown without restricting to a user. Only when this flag is set to 1 and a user is able to input an PIN number (password) correctly, showing a user the contents of that volume (reproduced) is permitted. When this flag is set to 0, even if a user does not input an PIN number, showing a user the contents of that volume is permitted.

[0119]First, when a user inserts a disk in a player. [whether this flag is set to 0, and] Or if a user is able to input an PIN number correctly even if this flag is set to 1, the recording and reproducing device 1 will display the list of PlayList in that disk. Reproduction restrictions of each PlayList are unrelated to volume_protect_flag, and it

is shown by playback_control_flag defined in UIAppInfoPlayList().

[0120]PIN comprises a number to four 0 thru/or 9, and each number is coded according to ISO/IEC646. The field of ref_thumbnail_index shows the information on the thumbnail image added to volume. In the case of the value whose ref_thumbnail_index field is not 0xFFFF, the thumbnail image is added to the volume and the thumbnail image is stored in a menu.thum file. The picture is referred to using the value of ref_thumbnail_index in a menu.thum file. When the ref_thumbnail_index field is 0xFFFF, it is shown that the thumbnail image is not added to the volume.

[0121]Next, TableOfPlayLists() in the syntax of info.dvr shown in drawing 15 is explained. TableOfPlayLists() stores the file name of PlayList (Real PlayList and Virtual PlayList). All the PlayList files currently recorded on volume are included in TableOfPlayList(). TableOfPlayLists() shows the default reproduction sequence of PlayList in volume.

[0122]Drawing 20 is a figure showing the syntax of TableOfPlayLists(), version_number of TableOfPlayLists shows four character characters which show the version number of this TableOfPlayLists for explaining that syntax. version_number must be coded with "0045" according to ISO 646.

[0123]length is an integer without 32-bit numerals which shows the number of bytes of TableOfPlayLists() from immediately after this length field to the last of TableOfPlayLists(). The 16-bit field of number_of_PlayLists shows the loop count of for-loop containing PlayList_file_name. This number must be equal to the number of PlayList(s) currently recorded on volume. 10 bytes of number of PlayList_file_name shows the file name of PlayList.

[0124]Drawing 21 is a figure showing the composition of another operation of the syntax of TableOfPlayLists(). The syntax shown in drawing 21 is considered as the composition which included UIAppinfoPlayList (after-mentioned) in the syntax shown in drawing 20. Thus, it becomes possible only by reading TableOfPlayLists to create a menu screen by having composition in which UIAppinfoPlayList was included. Here, the following explanation is given noting that the syntax shown in drawing 20 is used.

[0125]MakersPrivateData in the syntax of info.dvr shown in drawing 15 is explained. MakersPrivateData is provided so that the maker of the recording and reproducing device 1 can insert the private data of a maker into MakersPrivateData() for the special application of each company. The private data of each maker has maker_ID standardized in order to identify the maker which defined it. MakersPrivateData() may also contain one or more maker_ID.

[0126]When a predetermined maker wants to insert private data and the private data

of other makers is already contained in MakersPrivateData(), other makers, The old private data which already exists is not eliminated, but new private data is added into MakersPrivateData(). Thus, in this embodiment, the private data of two or more makers carries out as [be / being contained in one MakersPrivateData() / possible].

[0127]Drawing 22 is a figure showing the syntax of MakersPrivateData. version_number shows four character characters which show the version number of this MakersPrivateData() for explaining the syntax of MakersPrivateData shown in drawing 22. version_number must be coded with "0045" according to ISO 646. length shows the 32-bit unsigned integer which shows the number of bytes of MakersPrivateData() from immediately after this length field to the last of MakersPrivateData().

[0128]mpd_blocks_start_address shows the head byte address of the first mpd_block() by making the relative number of bytes from the byte of the head of MakersPrivateData() into a unit. A relative number of bytes is counted from zero. number_of_maker_entries is a 16-bit unsigned integer which gives the number of entries of the maker private data contained in MakersPrivateData(). Two or more maker private data which have a value of the same maker_ID in MakersPrivateData() must not exist.

[0129]mpd_block_size is a 16-bit unsigned integer which gives the size of one mpd_block by making 1024 bytes into a unit. For example, if it becomes mpd_block_size=1, it shows that the size of one mpd_block is 1024 bytes. number_of_mpd_blocks is a 16-bit unsigned integer which gives the number of mpd_block contained in MakersPrivateData(). maker_ID is a 16-bit unsigned integer which shows the manufacturing maker of the DVR system which created the maker private data. The value coded by maker_ID is specified by the licenser of this DVR format.

[0130]maker_model_code is a 16-bit unsigned integer which shows the model number code of the DVR system which created the maker private data. The value coded by maker_model_code is set up by the manufacturing maker who received the license of this format. start_mpd_block_number is a 16-bit unsigned integer which shows the number of mpd_block by which the maker private data is started. The aryne of the initial data of maker private data must be carried out to the head of mpd_block. start_mpd_block_number corresponds to the variable j in for-loop of mpd_block.

[0131]mpd_length is a 32-bit unsigned integer which shows the size of maker private data per byte. mpd_block is a field in which maker private data is stored. All the mpd_block in MakersPrivateData() must be the same sizes.

[0132]Next, xxxxx.rpls and yyyyy.vpls will be explained if it puts in another way about Real PlayList file and Virtual PlayList file. Drawing 23 is a figure showing the syntax of xxxxx.rpls (Real PlayList) or yyyyy.vpls (Virtual PlayList). xxxxx. rpls and yyyyy.vpls have the same syntax composition. xxxxx. rpls and yyyyy.vpls comprise three objects, respectively and they are PlayList(), PlayListMark(), and MakerPrivateData().

[0133]PlayListMark_Start_address shows the start address of PlayListMark() by making the relative number of bytes from the byte of the head of a PlayList file into a unit. A relative number of bytes is counted from zero.

[0134]MakerPrivateData_Start_address shows the start address of MakerPrivateData() by making the relative number of bytes from the byte of the head of a PlayList file into a unit. A relative number of bytes is counted from zero.

[0135]padding_word (padding word) is inserted according to the syntax of a PlayList file, and N1 and N2 are zero or arbitrary positive integers. It may be made for each padding word to take any value.

[0136]Here, although already explained simple, PlayList is explained further. Refer to the reproducing section in all the Clip(s) except Bridge-Clip (after-mentioned) for all the Real PlayList in a disk. And two or more RealPlayList(s) must not make the reproducing section shown by those PlayItem(s) overlap in the same Clip.

[0137]With reference to drawing 24, as shown, Real PlayList to which all the Clip(s) correspond exists in explaining further at drawing 24 (A). This rule is followed after editing work is performed, as shown in drawing 24 (B). therefore, all the Clip(s) -- which -- it is -- certainly viewing and listening is possible by referring to Real PlayList.

[0138]As shown in drawing 24 (C), the reproducing section of Virtual PlayList must be included in the reproducing section of RealPlayList, or the reproducing section of Bridge-Clip. Bridge-Clip referred to at no Virtual PlayList must not exist in a disk.

[0139]Although Real PlayList includes the list of PlayItem, it must not contain SubPlayItem. Virtual PlayList includes the list of PlayItem, CPI_type shown in PlayList() is EP_map type, And when PlayList_type is 0 (PlayList containing video and an audio), Virtual PlayList can contain one SubPlayItem. In PlayList() in this embodiment, SubPlayIte must be used only for the purpose of postrecording of an audio and the number of SubPlayItem(s) which one Virtual PlayList has must be 0 or 1.

[0140]Next, PlayList is explained. Drawing 25 is a figure showing the syntax of PlayList. They are four character characters in which version_number shows the version number of this PlayList() for explaining the syntax of PlayList shown in drawing 25. version_number must be coded with "0045" according to ISO 646. length is a 32-bit unsigned integer which shows the number of bytes of PlayList() from

immediately after this length field to the last of PlayList(). PlayList_type is the 8-bit field which shows the type of this PlayList, and shows drawing 26 that example.

[0141]CPI_type is a 1-bit flag and shows the value of CPI_type of Clip referred to by PlayItem() and SubPlayItem(). If all Clip(s) referred to by one PlayList do not have a the same value of CPI_type defined in those CPI(), they will not become. number_of_PlayItems is the 16-bit field which shows the number of PlayItem(s) in PlayList.

[0142]PlayItem_id corresponding to predetermined PlayItem() is defined by the turn that the PlayItem() appears, in for-loop containing PlayItem(). PlayItem_id is started from 0. number_of_SubPlayItems is the 16-bit field which shows the number of SubPlayItem(s) in PlayList. This value is 0 or 1. The path (audio stream path) of an additional audio stream is a kind of a sub path.

[0143]Next, UIAppInfoPlayList of the syntax of PlayList shown in drawing 25 is explained. UIAppInfoPlayList stores the parameter of the user interface application about PlayList. Drawing 27 is a figure showing the syntax of UIAppInfoPlayList. For explaining the syntax of UIAppInfoPlayList shown in drawing 27, character_set is the 8-bit field and shows the encoding method of the character character coded in the PlayList_name field. The encoding method corresponds to the value based on the table shown in drawing 19.

[0144]name_length is eight bit fields and shows the byte length of the PlayList name shown in the PlayList_name field. The field of PlayList_name shows the name of PlayList. The number of bytes of the left in this field to a name_length number is an effective character character, and it shows the name of PlayList. In the PlayList_name field, what kind of value may be [value after these effective character character] contained.

[0145]record_time_and_date is the 56-bit field in which time when PlayList is recorded is stored. This field codes 14 numbers by 4-bit Binary Coded Decimal (BCD) about a /part / second at the time of year / moon / day/. For example, 2001/12/23:01:02:03 are coded with "0x20011223010203."

[0146]duration is the 24-bit field which showed the total reproduction time of PlayList in the unit of time / part / second. This field codes six numbers by 4-bit Binary CodedDecimal (BCD). For example, 01:45:30 is coded with "0x014530."

[0147]valid_period is the 32-bit field which shows the period when PlayList is effective. This field codes eight numbers by 4-bit Binary Coded Decimal (BCD). For example, the recording and reproducing device 1 is used as it said that automatic deletion of the PlayList over which this shelf-life passed was carried out. For example,

2001/05/07 are coded with "0x20010507."

[0148]maker_id is a 16-bit unsigned integer which shows the manufacturer of the DVR player (recording and reproducing device 1) which updated the PlayList at the end. The value coded by maker_id is assigned by the licenser of a DVR format. maker_code is a 16-bit unsigned integer which shows the model number of the DVR player which updated the PlayList at the end. The value coded by maker_code is decided by the manufacturer who received the license of the DVR format.

[0149]The PlayList is reproduced, only when the flag of playback_control_flag is set to 1 and a user is able to input an PIN number correctly. When this flag is set to 0, even if a user does not input an PIN number, the user can view and listen to that PlayList.

[0150]As a table is shown write_protect_flag in drawing 28 (A), when being set to 1, write_protect_flag is removed, and the contents of the PlayList are not eliminated and changed. When this flag is set to 0, the user can eliminate and change that PlayList freely. When this flag is set to 1, before a user eliminates, edits or overwrites that PlayList, the recording and reproducing device 1 displays a message which is reconfirmed to a user.

[0151]Real PlayList by which write_protect_flag is set to zero exists, And Virtual PlayList which refers to Clip of the Real PlayList exists, and write_protect_flag of the Virtual PlayList may be set to one. When a user is going to eliminate RealPlayList, the recording and reproducing device 1, "Minimize" [it warns a user of existence of above-mentioned Virtual PlayList, or / the Real PlayList] before eliminating the Real PlayList.

[0152]As shown in drawing 28 (B), when the flag is set to 1, is_played_flag the PlayList, After being recorded, when having been reproduced is shown and it is set to 0 at once, the PlayList shows not being reproduced once, after being recorded.

[0153]archive is the 2-bit field which shows whether the PlayList is original or it is copied, as shown in drawing 28 (C). The field of ref_thumbnail_index shows the information on a thumbnail image that PlayList is represented. In the case of the value whose ref_thumbnail_index field is not 0xFFFF, the thumbnail image representing PlayList is added to the PlayList, and the thumbnail image is stored in a menu.thum file. The picture is referred to using the value of ref_thumbnail_index in a menu.thum file. When the ref_thumbnail_index field is 0xFFFF, the thumbnail image representing PlayList is not added to the PlayList.

[0154]Next, PlayItem is explained. One PlayItem() contains the following data fundamentally. Clip_information_file_name for specifying the file name of Clip, The pair of IN_time for pinpointing the reproducing section of Clip, and OUT_time, When

CPI_type defined in PlayList() is EP_map type, They are STC_sequence_id which IN_time and OUT_time refer to, and connection_condition which shows the state of connection between PlayItem to precede and the present PlayItem.

[0155]Those PlayItem(s) are arranged in on the global time-axis of PlayList without the gap of time, or overlap by the single tier when PlayList comprises two or more PlayItem(s). CPI_type defined in PlayList() is EP_map type, And the pair of IN_time defined in the PlayItem when the present PlayItem does not have BridgeSequence(), and OUT_time, The time on the same STC continuation section specified by STC_sequence_id must be pointed out. Such an example is shown in drawing 29.

[0156]CPI_type defined in PlayList() is EP_map type, and drawing 30 shows the case where the rule explained below is applied, when the present PlayItem has BridgeSequence(). IN_time (what is indicated to be IN_time1 in the figure) of PlayItem preceded with the present PlayItem has pointed out the time on the STC continuation section specified by STC_sequence_id of PlayItem to precede. OUT_time (what is indicated to be OUT_time1 in the figure) of PlayItem to precede has pointed out the time in Bridge-Clip specified in BridgeSequenceInfo() of the present PlayItem. This OUT_time must follow the coding restrictions mentioned later.

[0157]IN_time (what is indicated to be IN_time2 in the figure) of the present PlayItem has pointed out the time in Bridge-Clip specified in BridgeSequenceInfo() of the present PlayItem. This IN_time must also follow the coding restrictions mentioned later. OUT_time (what is indicated to be OUT_time2 in the figure) of PlayItem of the present PlayItem has pointed out the time on the STC continuation section specified by STC_sequence_id of the present PlayItem.

[0158]As shown in drawing 31, when CPI_type of PlayList() is TU_map type, the pair of IN_time of PlayItem and OUT_time has pointed out the time on the same Clip AV stream.

[0159]The syntax of PlayItem comes to be shown in drawing 32. The field of Clip_Information_file_name shows the file name of ClipInformation file for explaining the syntax of PlayItem shown in drawing 32. Clip_stream_type defined in ClipInfo() of this Clip Information file must show Clip AV stream.

[0160]STC_sequence_id is the 8-bit field and shows STC_sequence_id of the STC continuation section which PlayItem refers to. When CPI_type specified in PlayList() is TU_map type, these eight bit fields have no meanings, but are set to 0. IN_time is 32 bit fields and stores the reproduction start time of PlayItem. The semantics of IN_time changes with CPI_type defined in PlayList(), as shown in drawing 33.

[0161]OUT_time is 32 bit fields and stores the reproduction finish time of PlayItem.

The semantics of OUT_time changes with CPI_type defined in PlayList(), as shown in drawing 34.

[0162]Connection_Condition is the 2-bit field which shows the connected state between PlayItem to precede as shown in drawing 35, and the present PlayItem. Drawing 36 is a figure explaining each state of Connection_Condition shown in drawing 35.

[0163]Next, BridgeSequenceInfo is explained with reference to drawing 37. BridgeSequenceInfo() is the attached information of the present PlayItem and has the information shown below. Bridge_Clip_Information_file_name which specifies a Bridge-Clip AV stream file and ClipInformation file corresponding to it is included.

[0164]It is an address of the source packet on Clip AV stream which PlayItem to precede refers to, and the source packet of the beginning of a Bridge-Clip AV stream file is connected following this source packet. This address is called RSPN_exit_from_previous_Clip. It is an address of the source packet on Clip AV stream which the further present PlayItem refers to, and the source packet of the last of a Bridge-Clip AV stream file is connected before this source packet. This address is called RSPN_enter_to_current_Clip.

[0165]In drawing 37, RSPN_arrival_time_discontinuity shows the address of the source packet which has a break point of arrival time base in a the Bridge-Clip AVstream file. This address is defined in ClipInfo().

[0166]Drawing 38 is a figure showing the syntax of BridgeSequenceinfo. The syntax of BridgeSequenceinfo shown in drawing 38 to explain the field of Bridge_Clip_Information_file_name, The file name of Clip Information file corresponding to a Bridge-Clip AV stream file is shown. Clip_stream_type defined in ClipInfo() of this Clip Information file must show 'Bridge-Clip AV stream'.

[0167]32 bit fields of RSPN_exit_from_previous_Clip, It is a relative address of the source packet on Clip AV stream which PlayItem to precede refers to, and the source packet of the beginning of a Bridge-Clip AV stream file is connected following this source packet. RSPN_exit_from_previous_Clip, It is a size which makes a source packet number a unit, and the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of the Clip AV stream file which PlayItem to precede refers to is counted as an initial value.

[0168]32 bit fields of RSPN_enter_to_current_Clip, It is a relative address of the source packet on Clip AV stream which the present PlayItem refers to, and the source packet of the last of a Bridge-Clip AV stream file is connected before this source packet. RSPN_exit_from_previous_Clip, It is a size which makes a source packet

number a unit, and the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of the Clip AV stream file which the present PlayItem refers to is counted as an initial value.

[0169]Next, SubPlayItem is explained with reference to drawing 39. Use of SubPlayItem() is allowed only when CPI_type of PlayList() is EP_map type. In this embodiment, SubPlayItem presupposes that it is used only for the purpose of postrecording of an audio. SubPlayItem() contains the data shown below. First, Clip_information_file_name for specifying Clip which sub path in PlayList refers to is included.

[0170]SubPath_IN_time and SubPath_OUT_time for specifying the reproducing section of sub path in Clip are included. sync_PlayItem_id and sync_start_PTS_of_PlayItem for specifying the time in which sub path carries out a reproduction start on the time-axis of main path are included. Clip AV stream of the audio referred to at sub path must not contain an STC break point (break point of system time base). The clock of the audio sample of Clip used for sub path is locked by the clock of the audio sample of main path.

[0171]Drawing 40 is a figure showing the syntax of SubPlayItem. The syntax of SubPlayItem shown in drawing 40 to explain the field of Clip_Information_file_name, The file name of Clip Information file is shown and it is used by sub path in PlayList. Clip_stream_type defined in ClipInfo() of this Clip Information file must show Clip AV stream.

[0172]The 8-bit field of SubPath_type shows the type of sub path. Here, as shown in drawing 41, only '0x00' is set up but other values are secured for the future.

[0173]The 8-bit field of sync_PlayItem_id shows PlayItem_id of PlayItem in which the time in which sub path carries out a reproduction start on the time-axis of main path is contained. The value of PlayItem_id corresponding to predetermined PlayItem is defined in PlayList() (refer to drawing 25).

[0174]The 32-bit field of sync_start_PTS_of_PlayItem, The time in which sub path carries out a reproduction start on the time-axis of main path is shown, and top 32 bits of PTS (Presentaiotn Time Stamp) on PlayItem referred to by sync_PlayItem_id are shown. 32 bit fields of SubPath_IN_time store the reproduction start time of Sub path. SubPath_IN_time shows top 32 bits of PTS of 33 bit length corresponding to the first presentation unit in Sub Path.

[0175]32 bit fields of SubPath_OUT_time store the reproduction finish time of Sub path. SubPath_OUT_time shows top 32 bits of the value of Presenation_end_TS computed by a following formula. Presentation_end_TS = PTS_out+AU_duration --

here, PTS_out is PTS of 33 bit length corresponding to the presentation unit of the last of SubPath. AU_duration is a display period of the 90-kHz unit of the presentation unit of the last of SubPath.

[0176]Next, PlayListMark() in the syntax of xxxxx.rpls shown in drawing 23 and yyyyy.vpls is explained. The mark information about PlayList is stored in this PlayListMark. Drawing 42 is a figure showing the syntax of PlayListMark. They are four character characters in which version_number shows the version number of this PlayListMark() for explaining the syntax of PlayListMark shown in drawing 42. version_number must be coded with "0045" according to ISO 646.

[0177]length is a 32-bit unsigned integer which shows the number of bytes of PlayListMark() from immediately after this length field to the last of PlayListMark(). number_of_PlayList_marks is a 16-bit unsigned integer which shows the number of the mark currently stored in PlayListMark. number_of_PlayList_marks may be 0. mark_type is the 8-bit field which shows the type of a mark, and is coded according to the table shown in drawing 43.

[0178]32 bit fields of mark_time_stamp store the time stamp in which the point as which the mark was specified is shown. The semantics of mark_time_stamp changes with CPI_type defined in PlayList(), as shown in drawing 44. PlayItem_id is the 8-bit field which specifies PlayItem on which the mark is put. The value of PlayItem_id corresponding to predetermined PlayItem is defined in PlayList() (refer to drawing 25).

[0179]The 8-bit field of character_set shows the encoding method of the character character coded by the mark_name field. The encoding method corresponds to the value shown in drawing 19. Eight bit fields of name_length show the byte length of the mark name shown in the Mark_name field. The field of mark_name shows the name of a mark. The number of bytes of the left in this field to a name_length number is an effective character character, and it shows the name of a mark. As for the value after these effective character character, what kind of value may be set up in the Mark_name field.

[0180]The field of ref_thumbnail_index shows the information on the thumbnail image added to a mark. In the case of the value whose ref_thumbnail_index field is not 0xFFFF, the thumbnail image is added to the mark and the thumbnail image is stored in a mark.thmb file. The picture is referred to using the value of ref_thumbnail_index in a mark.thmb file (after-mentioned). When the ref_thumbnail_index field is 0xFFFF, it is shown that the thumbnail image is not added to the mark.

[0181]Next, Clip information file is explained. zzzzz.clpi (Clip information file file) comprises six objects, as shown in drawing 45. They are ClipInfo(), STC_Info(),

ProgramInfo(), CPI(), ClipMark(), and MakerPrivateData(). "zzzzz" of the digit string with same AV stream (Clip AV stream or Bridge-Clip AV stream) and Clip Information file corresponding to it is used.

[0182]to explain the syntax of zzzzz.clpi (Clip information file file) shown in drawing 45, ClipInfo_Start_address shows the start address of ClipInfo() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero.

[0183]STC_Info_Start_address shows the start address of STC_Info() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. ProgramInfo_Start_address shows the start address of ProgramInfo() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. CPI_Start_address shows the start address of CPI() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero.

[0184]ClipMark_Start_address shows the start address of ClipMark() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. MakerPrivateData_Start_address shows the start address of MakerPrivateData () by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. padding_word (padding word) is inserted according to the syntax of a zzzzz.clpi file. N1, N2, N3, N4, and N5 must be zero or arbitrary positive integers. Any value may be made to be taken each padding word.

[0185]Next, ClipInfo is explained. Drawing 46 is a figure showing the syntax of ClipInfo. ClipInfo() stores the attribution information of the AV stream file (a Clip AV stream or a Bridge-Clip AV stream file) corresponding to it.

[0186]They are four character characters in which version_number shows the version number of this ClipInfo() for explaining the syntax of ClipInfo shown in drawing 46. version_number must be coded with "0045" according to ISO 646. length is a 32-bit unsigned integer which shows the number of bytes of ClipInfo() from immediately after this length field to the last of ClipInfo(). The 8-bit field of Clip_stream_type shows the type of the AV stream corresponding to a Clip Information file, as shown in drawing 47. The stream type of each type of AV stream is mentioned later.

[0187]The 32-bit field of offset_SPN gives the offset value of the source packet number about the source packet of the beginning of an AV stream (Clip AV stream or Bridge-Clip AV stream) file. This offset_SPN must be 0 when an AV stream file is first

recorded on a disk.

[0188]As shown in drawing 48, when the first portion of an AV stream file is eliminated by edit, offset_SPN is very good in values other than zero. The relative source packet number (relative address) which refers to offset_SPN in this embodiment is often RSPN_xxx (xxx changes.). It is described by in the form of example .RSPN_EP_start in syntax. A relative source packet number is a size which makes a source packet number a unit, and counts the value of offset_SPN as an initial value from the source packet of the beginning of an AV stream file.

[0189]The number (SPN_xxx) of the source packets to the source packet referred to by a relative source packet number from the source packet of the beginning of an AV stream file is computed with a following formula.

offset_SPN shows SPN_xxx = RSPN_xxx - offset_SPN drawing 48 the example in the case of being 4.

[0190]TS_recording_rate is a 24-bit unsigned integer -- this value -- a DVR drive (writing part 22) -- or the bit rate of required input and output of the AV stream from a DVR drive (read section 28) is given. record_time_and_date, It is the 56-bit field in which time when the AV stream corresponding to Clip is recorded is stored, and 14 numbers are coded by 4-bit Binary Coded Decimal (BCD) about a /part / second at the time of year / moon / day/. For example, 2001/12/23:01:02:03 are coded with "0x20011223010203."

[0191]duration is the 24-bit field which showed the total reproduction time of Clip in the unit of time / part / second based on an arrival time clock. This field codes six numbers by 4-bit Binary Coded Decimal (BCD). For example, 01:45:30 is coded with "0x014530."

[0192]The flag of time_controlled_flag: shows the recording mode of an AV stream file. When this time_controlled_flag is 1, the recording mode must fulfill the conditions which show that it is the mode in which it is recorded to the time progress after recording as a file size is proportional, and are shown in a following formula.

TS_average_rate*192/188*(t - start_time)-a <= size_clip -- (t) <= TS_average_rate*192 / 188*(t - start_time)+alpha -- here, TS_average_rate expresses the average bit rate of the transport stream of an AV stream file with a bytes/second unit.

[0193]In an upper type, t shows the time expressed with a second bit, and start_time is time when the source packet of the beginning of an AV stream file is recorded, and is expressed with a second bit. When the size of size_clip (t) and the AV stream file in the time t is expressed per byte, for example, ten source packets are recorded by the time t from start_time, size_clip (t) is 10*192 byte. a is a constant depending on

TS_average_rate.

[0194]When time_controlled_flag is set to zero, not controlling the recording mode so that the file size of an AV stream is proportional to time progress of record is shown. For example, this is a case where transparent record of the input transport stream is carried out.

[0195]When, as for TS_average_rate, time_controlled_flag is set to one, this 24-bit field shows the value of TS_average_rate used by the upper formula. When time_controlled_flag is set to zero, this field has no meanings but must be set to 0. For example, the transport stream of a Variable Bit Rate is coded by the procedure shown below. A transformer portrait is first set to the value of TS_recording_rate. Next, a video stream is coded with a Variable Bit Rate. And a transport packet is intermittently coded by not using null packets.

[0196]32 bit fields of RSPN_arrival_time_discontinuity are the relative addresses of the place which the discontinuity of arrival time base generates on a Bridge-ClipAV stream file. RSPN_arrival_time_discontinuity, It is a size which makes a source packet number a unit, and the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of a Bridge-Clip AV stream file is counted as an initial value. The absolute address in the inside of the Bridge-Clip AV stream file is computed based on SPN_xxx = RSPN_xxx - offset_SPN mentioned above.

[0197]The 144-bit field of reserved_for_system_use is reserved for systems. When the flag of is_format_identifier_valid is 1, it is shown that the field of format_identifier is effective. When the flag of is_original_network_ID_valid is 1, it is shown that the field of original_network_ID is effective. When the flag of is_transport_stream_ID_valid is 1, it is shown that the field of transport_stream_ID is effective. When the flag of is_servece_ID_valid is 1, it is shown that the field of servece_ID is effective.

[0198]When the flag of is_country_code_valid is 1, it is shown that the field of country_code is effective. 32 bit fields of format_identifier show the value of format_identifier which registration deascriotor (it defines as ISO/IEC 13818-1) has in a transport stream. 16 bit fields of original_network_ID show the value of original_network_ID defined in the transport stream. 16 bit fields of transport_stream_ID show the value of transport_stream_ID defined in the transport stream.

[0199]16 bit fields of servece_ID show the value of servece_ID defined in the transport stream. The 24-bit field of country_code shows the country code defined by ISO3166. Each character character is coded by ISO8859-1. For example, Japan is expressed as "JPN" and coded with "0x4A 0x500x4E." stream_format_name is 16 character codes

of ISO-646 which shows the name of the format organization which is carrying out the stream definition of the transport stream. The invalid byte in this field, value'0xFF' is set.

[0200]format_identifier, original_network_ID, transport_stream_ID, servece_ID, country_code, and stream_format_name, The service provider of a transport stream is shown and by this, Coding restrictions of an audio or a video stream and the stream definition of the standard of SI (service information) or private data streams other than an audio video stream can be recognized. These information can be used, in order that a decoder may perform initial setting of a decoder system before a decoding start whether the stream can be decoded and when it can decode and.

[0201]Next, STC_Info is explained. In MPEG-2 transport stream, here call STC_sequence the time intervals which do not contain the break point (break point of system time base) of STC, and in Clip, STC_sequence is specified with the value of STC_sequence_id. Drawing 50 is a figure explaining the STC section [****]. The value of the STC same in the same STC_sequence never appears (however, the maximum time length of Clip is restricted so that it may mention later). Therefore, the value of the same PTS also never appears in the same STC_sequence. When an AV stream contains the STC break point of N (N> 0) individual, the system time base of Clip is divided into STC_sequence of an individual (N+1).

[0202]STC_Info stores the address of the place which the discontinuity (discontinuity of system time base) of STC generates. So that it may explain with reference to drawing 51 RSPN_STC_start, The address is shown and k-th STC_sequence (k>=0) except the last STC_sequence, It begins from the time when the source packet referred to by k-th RSPN_STC_start arrived, and finishes with the time when the source packet referred to by RSPN_STC_start of eye watch (k+1) arrived. The last STC_sequence begins from the time when the source packet referred to by the last RSPN_STC_start arrived, and is ended at the time when the last source packet arrived.

[0203]Drawing 52 is a figure showing the syntax of STC_Info. They are four character characters in which version_number shows the version number of this STC_Info() for explaining the syntax of STC_Info shown in drawing 52. version_number must be coded with "0045" according to ISO 646.

[0204]length is a 32-bit unsigned integer which shows the number of bytes of STC_Info() from immediately after this length field to the last of STC_Info(). When CPI_type of CPI() shows TU_map type, this length field may set zero. When CPI_type of CPI() shows EP_map type, num_of_STC_sequences must be one or more values.

[0205]The 8-bit unsigned integer of num_of_STC_sequences shows the number of

STC_sequence in the inside of Clip. This value shows the loop count of for-loop following this field. STC_sequence_id corresponding to predetermined STC_sequence is defined in for-loop containing RSPN_STC_start by the turn that RSPN_STC_start corresponding to the STC_sequence appears. STC_sequence_id is started from 0.

[0206]32 bit fields of RSPN_STC_start show the address which STC_sequence starts on an AV stream file. RSPN_STC_start shows the address which the break point of system time base generates in an AV stream file. RSPN_STC_start is good also as a relative address of a source packet which has PCR of the beginning of new system time base in an AV stream. RSPN_STC_start is a size which makes a source packet number a unit, and counts the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by SPN_xxx = RSPN_xxx - offset_SPN already mentioned above.

[0207]Next, ProgramInfo in the syntax of zzzzz.clip shown in drawing 45 is explained. The time intervals which have the following feature in Clip are called program_sequence for explaining here, referring to drawing 53. First, the value of PCR_PID does not change. Next, the number of video elementary streams does not change. The encoded information defined by the value and VideoCodingInfo of PID about each video stream does not change. The number of audio elementary streams does not change. The encoded information defined by the value and AudioCodingInfo of PID about each audio stream does not change.

[0208]In the same time, program_sequence has only one system time base. In the same time, program_sequence has only one PMT. ProgramInfo() stores the address of the place which program_sequence starts. RSPN_program_sequence_start shows the address.

[0209]Drawing 54 is a figure showing the syntax of ProgramInfo. They are four character characters in which version_number shows the version number of this ProgramInfo() for explaining SHINTAKU of ProgramInfo shown in drawing 54. version_number must be coded with "0045" according to ISO 646.

[0210]length is a 32-bit unsigned integer which shows the number of bytes of ProgramInfo() from immediately after this length field to the last of ProgramInfo(). When CPI_type of CPI() shows TU_map type, this length field may be set to zero. When CPI_type of CPI() shows EP_map type, number_of_programs must be one or more values.

[0211]The 8-bit unsigned integer of number_of_program_sequences shows the number of program_sequence in the inside of Clip. This value shows the loop count of for-loop

following this field. When program_sequence does not change in Clip, number_of_program_sequences must have one set. 32 bit fields of RSPN_program_sequence_start are the relative addresses of the place which a program sequence starts on an AV stream file.

[0212]RSPN_program_sequence_start is a size which makes a source packet number a unit, and counts the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by SPN_xxx = RSPN_xxx - offset_SPN. The RSPN_program_sequence_start value must appear in an ascending order in for-loop of syntax.

[0213]16 bit fields of PCR_PID show PID of a transport packet including the PCR field effective in the program_sequence. Eight bit fields of number_of_videos show the loop count of video_stream_PID and for-loop containing VideoCodingInfo(). Eight bit fields of number_of_audios show the loop count of audio_stream_PID and for-loop containing AudioCodingInfo(). 16 bit fields of video_stream_PID show PID of the transport packet containing a video stream effective in the program_sequence. VideoCodingInfo() following this field must explain the contents of the video stream referred to by that video_stream_PID.

[0214]16 bit fields of audio_stream_PID show PID of the transport packet containing an audio stream effective in the program_sequence. AudioCodingInfo() following this field must explain the contents of the video stream referred to by that audio_stream_PID.

[0215]The turn that the value of video_stream_PID appears in for-loop of syntax must be equal to the turn that PID of the video stream is coded in PMT effective in the program_sequence. The turn that the value of audio_stream_PID appears in for-loop of syntax must be equal to the turn that PID of the audio stream is coded in PMT effective in the program_sequence.

[0216]Drawing 55 is a figure showing the syntax of VideoCodingInfo in the syntax of Programinfo shown in drawing 54. For explaining the syntax of VideoCodingInfo shown in drawing 55, eight bit fields of video_format show the format video corresponding to video_stream_PID in ProgramInfo(), as shown in drawing 56.

[0217]Eight bit fields of frame_rate show the frame rate of the video corresponding to video_stream_PID in ProgramInfo(), as shown in drawing 57. Eight bit fields of display_aspect_ratio show the display aspect ratio of the video corresponding to video_stream_PID in ProgramInfo(), as shown in drawing 58.

[0218]Drawing 59 is a figure showing the syntax of AudioCodingInfo in the syntax of

Programinfo shown in drawing 54. For explaining the syntax of AudioCodingInfo shown in drawing 59, eight bit fields of audio_coding show the encoding method of the audio corresponding to audio_stream_PID in ProgramInfo(), as shown in drawing 60.

[0219]Eight bit fields of audio_component_type show the component type of the audio corresponding to audio_stream_PID in ProgramInfo(), as shown in drawing 61. Eight bit fields of sampling_frequency show the sampling frequency of the audio corresponding to audio_stream_PID in ProgramInfo(), as shown in drawing 62.

[0220]Next, CPI (Characteristic Point Information) in the syntax of zzzzz.clip shown in drawing 45 is explained. Since the hour entry in an AV stream and the address in the file are associated, there is CPI. There are two types of CPI(s) and they are EP_map and TU_map. As shown in drawing 63, when CPI_type in CPI() is EP_map type, the CPI() contains EP_map. As shown in drawing 64, when CPI_type in CPI() is TU_map type, the CPI() contains TU_map. One AV stream has one EP_map or one TU_map. When an AV stream is a SESF transport stream, Clip corresponding to it must have EP_map.

[0221]Drawing 65 is a figure showing the syntax of CPI. They are four character characters in which version_number shows the version number of this CPI() for explaining the syntax of CPI shown in drawing 65. version_number must be coded with "0045" according to ISO 646. length is a 32-bit unsigned integer which shows the number of bytes of CPI() from immediately after this length field to the last of CPI(). As shown in drawing 66, CPI_type is a 1-bit flag and expresses the type of CPI of Clip.

[0222]Next, EP_map in the syntax of CPI shown in drawing 65 is explained. There are two types of EP_map and it is EP_map for video streams, and EP_map for audio streams. EP_map_type in EP_map distinguishes the type of EP_map. When Clip contains one or more video streams, EP_map for video streams must be used. When Clip does not contain a video stream but contains one or more audio streams, EP_map for audio streams must be used.

[0223]EP_map for video streams is explained with reference to drawing 67. EP_map for video streams has data called stream_PID, PTS_EP_start, and RSPN_EP_start. stream_PID shows PID of the transport packet which transmits a video stream. PTS_EP_start shows PTS of the access unit begun from the sequence header of a video stream. RSPN_EP_start shows the address of the sauce pocket containing the 1st byte of the access unit referred to by PTS_EP_start in an AV stream.

[0224]The sub table called EP_map_for_one_stream_PID() is made for every video stream transmitted by the transport packet with the same PID. When two or more video streams exist in Clip, EP_map may also contain two or more

EP_map_for_one_stream_PID().

[0225]EP_map for audio streams has data called stream_PID, PTS_EP_start, and RSPN_EP_start. stream_PID shows PID of the transport packet which transmits an audio stream. PTS_EP_start shows PTS of the access unit of an audio stream. RSPN_EP_start shows the address of the sauce pocket containing the 1st byte of the access unit referred to by PTS_EP_start in an AV stream.

[0226]The sub table called EP_map_for_one_stream_PID() is made for every audio stream transmitted by the transport packet with the same PID. When two or more audio streams exist in Clip, EP_map may also contain two or more EP_map_for_one_stream_PID().

[0227]One EP_map_for_one_stream_PID() is made by explaining the relation between EP_map and STC_Info on one table regardless of the break point of STC. By comparing the value of RSPN_STC_start defined in the value of RSPN_EP_start and STC_Info() shows the boundary of the data of EP_map belonging to each STC_sequence (see drawing 68). - EP_map must have one EP_map_for_one_stream_PID to the range of the continuous stream transmitted by the same PID. When shown in drawing 69, although program#1 and program#3 have the same video PID, since the data range is not continuing, they must have EP_map_for_one_stream_PID for every program.

[0228]Drawing 70 is a figure showing the syntax of EP_map. For explaining the syntax of EP_map shown in drawing 70, EP_type is the 4-bit field, and as shown in drawing 71, it shows the entry point type of EP_map. EP_type shows the semantics of the data field following this field. EP_type must be set to zero ('video') when Clip contains one or more video streams. Or EP_type must be set to one ('audio'), when Clip does not contain a video stream but contains one or more audio streams.

[0229]The 16-bit field of number_of_stream_PIDs shows the loop count of for-loop which has number_of_stream_PIDs in EP_map() in a variable. The 16-bit field of stream_PID (k), PID of the transport packet which transmits the k-th elementary stream (video or audio stream) referred to by EP_map_for_one_stream_PID (num_EP_entries (k)) is shown. case EP_type is equal to zero ('video') -- the elementalist ream -- a video stream -- kicking does not become impossible. When EP_type is equal to one ('audio'), the elementalist ream must be an audio stream.

[0230]The 16-bit field of num_EP_entries (k) shows num_EP_entries (k) referred to by EP_map_for_one_stream_PID (num_EP_entries (k)).

EP_map_for_one_stream_PID_Start_address (k): This 32-bit field, The relative byte position from which EP_map_for_one_stream_PID (num_EP_entries (k)) begins in EP_map() is shown. This value is shown by the size from the 1st byte of EP_map().

[0231]padding_word must be inserted according to the syntax of EP_map(). X and Y must be zero or arbitrary positive integers. Each padding word may take any value.

[0232]Drawing 72 is a figure showing the syntax of EP_map_for_one_stream_PID. The semantics of the 32-bit field of PTS_EP_start changes with EP_type defined in EP_map() to explain the syntax of EP_map_for_one_stream_PID shown in drawing 72. When EP_type is equal to zero ('video'), this field has top 32 bits of PTS of the 33-bit accuracy of the access unit which starts with the sequence header of a video stream. When EP_type is equal to one ('audio'), this field has top 32 bits of PTS of the 33-bit accuracy of the access unit of an audio stream.

[0233]The semantics of the 32-bit field of RSPN_EP_start changes with EP_type defined in EP_map(). When EP_type is equal to zero ('video'), this field shows the relative address of the sauce pocket containing the 1st byte of the sequence header of the access unit referred to by PTS_EP_start in an AV stream. Or when EP_type is equal to one ('audio'), this field shows the relative address of the sauce pocket containing the first byte of the audio frame of the access unit referred to by PTS_EP_start in an AV stream.

[0234]RSPN_EP_start is a size which makes a source packet number a unit, and counts the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by SPN_xxx = RSPN_xxx - offset_SPN. The value of RSPN_EP_start must appear in an ascending order in for-loop of syntax.

[0235]Next, TU_map is explained with reference to drawing 73. TU_map makes one time-axis based on the arrival time clock (clock of an arrival time base) of a source packet. The time-axis is called TU_map_time_axis. The starting point of TU_map_time_axis is shown by offset_time in TU_map(). TU_map_time_axis is divided into a fixed unit from offset_time. The unit is called time_unit.

[0236]In each time_unit in an AV stream, the address on the AV stream file of the source packet of the first perfect form is stored in TU_map. These addresses are called RSPN_time_unit_start. In a TU_map_time_axis top, it is k. The time when time_unit of eye watch (k>=0) begins is called TU_start_time (k). This value is computed based on a following formula.

TU_start_time (k) = offset_time+k*time_unit_sizeTU_start_time (k) has the accuracy of 45 kHz.

[0237]Drawing 75 is a figure showing the syntax of TU_map. The field of the 32-bit length of offset_time gives the offset time to TU_map_time_axis for explaining the syntax of TU_map shown in drawing 75. This value shows the offset time to time_unit

of the beginning in Clip. offset_time is a size which makes a unit the 45-kHz clock drawn from the arrival time clock of 27-MHz accuracy. offset_time must be set to zero when an AV stream is recorded as new Clip.

[0238]32 bit fields of time_unit_size give the size of time_unit, and it is a size which makes a unit the 45-kHz clock drawn from the arrival time clock of 27-MHz accuracy. time_unit_size is good to use 1 or less (time_unit_size<=45000) second. 32 bit fields of number_of_time_unit_entries show the number of entries of time_unit currently stored in TU_map().

[0239]32 bit fields of RSPN_time_unit_start show the relative address of the place which each time_unit starts in an AV stream. RSPN_time_unit_start is a size which makes a source packet number a unit, and counts the value of offset_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by SPN_xxx = RSPN_xxx - offset_SPN. The value of RSPN_time_unit_start must appear in an ascending order in for-loop of syntax. (k+1) When anything does not have a source packet in time_unit of eye watch, RSPN_time_unit_start of eye watch (k+1) must be equal to k-th RSPN_time_unit_start.

[0240]ClipMark in the syntax of zzzzz.clip shown in drawing 45 is explained. ClipMark is the mark information about a clip and is stored into ClipMark. This mark is set by a recorder (recording and reproducing device 1), and is not set by the user.

[0241]Drawing 75 is a figure showing the syntax of ClipMark. They are four character characters in which version_number shows the version number of this ClipMark() for explaining the syntax of ClipMark shown in drawing 75. version_number must be coded with "0045" according to ISO 646.

[0242]length is a 32-bit unsigned integer which shows the number of bytes of ClipMark() from immediately after this length field to the last of ClipMark(). The 16-bit unsigned integer number_of_Clip_marks indicates the number of the mark currently stored in ClipMark to be. number_of_Clip_marks may be 0. mark_type is the 8-bit field which shows the type of a mark, and is coded according to the table shown in drawing 76.

[0243]mark_time_stamp is 32 bit fields and stores the time stamp in which the point as which the mark was specified is shown. The semantics of mark_time_stamp changes with CPI_type in PlayList(), as shown in drawing 77.

[0244]When, as for STC_sequence_id, CPI_type in CPI() shows EP_map type, this 8-bit field shows STC_sequence_id of the STC continuation section on which the mark is put. When CPI_type in CPI() shows TU_map type, this 8-bit field has no meanings, but

is set to zero. The 8-bit field of character_set shows the encoding method of the character character coded by the mark_name field. The encoding method corresponds to the value shown in drawing 19.

[0245]Eight bit fields of name_length show the byte length of the mark name shown in the Mark_name field. The field of mark_name shows the name of a mark. The number of bytes of the left in this field to a name_length number is an effective character character, and it shows the name of a mark. In the mark_name field, what kind of value may be [value after these effective character character] contained.

[0246]The field of ref_thumbnail_index shows the information on the thumbnail image added to a mark. In the case of the value whose ref_thumbnail_index field is not 0xFFFF, the thumbnail image is added to the mark and the thumbnail image is stored in a mark.thmb file. The picture is referred to using the value of ref_thumbnail_index in a mark.thmb file. When the ref_thumbnail_index field is 0xFFFF, the thumbnail image is not added to the mark.

[0247]Since MakersPrivateData was already explained with reference to drawing 22, the explanation is omitted.

[0248]Next, the thumbnail information (Thumbnail Information) is explained. A thumbnail image is stored in a menu.thmb file or a mark.thmb file. These files are the same syntax structures and have only one Thumbnail(). A menu.thmb file stores a menu thumbnail image, i.e., the picture representing Volume, and the picture representing each PlayList. All the menu thumbnails are stored only in one menu.thmb file.

[0249]A mark.thmb file stores the picture showing a mark thumbnail image, i.e., a marking point. All the mark thumbnails to all the PlayList(s) and Clip(s) are stored only in one mark.thmb file. Since a thumbnail is added and deleted frequently, add operation and operation of partial deletion must be able to be performed at high speed easily. Thumbnail() has a block structure for this reason. The data of a picture is divided into some portions and each portion is stored in one tn_block. One image data is stored in tn_block which ******(ed). tn_block which is not used may exist in the sequence of tn_block. The byte length of one thumbnail image is variable.

[0250]Drawing 78 is a figure showing the syntax of menu.thmb and mark.thmb, and drawing 79 is a figure showing the syntax of Thumbnail in the syntax of menu.thmb shown in drawing 78, and mark.thmb. They are four character characters in which version_number shows the version number of this Thumbnail() for explaining the syntax of Thumbnail shown in drawing 79. version_number must be coded with "0045" according to ISO 646.

[0251]length is a 32-bit unsigned integer which shows the number of bytes of MakersPrivateData() from immediately after this length field to the last of Thumbnail(). tn_blocks_start_address is a 32-bit unsigned integer which shows the head byte address of the first tn_block by making the relative number of bytes from the byte of the head of Thumbnail() into a unit. A relative number of bytes is counted from zero. number_of_thumbnails is a 16-bit unsigned integer which gives the number of entries of the thumbnail image contained in Thumbnail().

[0252]tn_block_size is a 16-bit unsigned integer which gives the size of one tn_block by making 1024 bytes into a unit. For example, if it becomes tn_block_size=1, it shows that the size of one tn_block is 1024 bytes. number_of_tn_blocks is a 116-bit unsigned integer showing the number of entries of tn_block in this Thumbnail(). thumbnail_index is a 16-bit unsigned integer showing the index number of a thumbnail image expressed with the thumbnail information on the "for" loop batch which begins from this thumbnail_index field. Don't use a value called 0xFFFF as thumbnail_index. Refer to thumbnail_index for ref_thumbnail_index in UIAppInfoVolume(), UIAppInfoPlayList(), PlayListMark(), and ClipMark().

[0253]thumbnail_picture_format is an 8-bit unsigned integer showing the picture format of a thumbnail image, and takes a value as shown in drawing 80. DCF and PNG in front are allowed only within "menu.thmb." The mark thumbnail must take value"0x00" (MPEG-2 Video I-picture).

[0254]picture_data_size is a 32-bit unsigned integer which shows the byte length of a thumbnail image per byte. start_tn_block_number is a 16-bit unsigned integer showing the tn_block number of tn_block from which the data of a thumbnail image begins. The head of thumbnail image data must be in agreement with the head of tb_block. A tn_block number begins from 0 and is related to the value of the variable k in the for-loop of tn_block.

[0255]x_picture_length is a 16-bit unsigned integer showing the horizontal number of pixels of the frame picture frame of a thumbnail image. y_picture_length is a 16-bit unsigned integer showing the number of pixels of the perpendicular direction of the frame picture frame of a thumbnail image. tn_block is a field in which a thumbnail image is stored. All the tn_block in Thumbnail() is the same sizes (fixed length), and the size is defined by tn_block_size.

[0256]Drawing 81 is the figure which meant typically how thumbnail image data would be stored in tn_block. Like drawing 81, each thumbnail image data begins from the head of tn_block, and, in the case of the size exceeding 1 tn_block, it is stored using continuous following tn_block. By doing in this way, the picture data which is variable

length becomes possible [managing as fixed-length data], and can respond now by simple processing to edit called deletion.

[0257]Next, an AV stream file is explained. An AV stream file is stored in an "M2TS" directory (drawing 14). There are two types of AV stream files, and they are a Clip AV stream and a Bridge-Clip AV stream file. It must be the structure of a DVR MPEG-2 transport stream file where both AV streams are defined henceforth [this].

[0258]First, DVR MPEG-2 A transport stream is explained. The structure of DVR MPEG-2 transport stream is shown in drawing 82. An AV stream file has the structure of a DVR MPEG2 transport stream. A DVR MPEG2 transport stream comprises Aligned unit of integer pieces. The size of Alignedunit is 6144. Byte (2048*3 byte) It is. Aligned unit begins from the 1st byte of a source packet. A source packet is 192-byte length. One source packet comprises TP_extra_header and a transport packet. TP_extra_header is 4-byte length and a transport packet is 188-byte length.

[0259]One Aligned unit comprises 32 source packets. Aligned unit of the last in a DVR MPEG2 transport stream also comprises 32 source packets. Therefore, the termination of the DVR MPEG2 transport stream is carried out on the boundary of Aligned unit. When the number of the transport packets of the input transport stream recorded on a disk is not a multiple of 32, a source packet with null packets (transport packet of PID=0x1FFF) must be used for the last Aligned unit. The file system must not add excessive information to a DVR MPEG2 transport stream.

[0260]The recorder model of DVR MPEG-2 transport stream is shown in drawing 83. The recorder shown in drawing 83 is a model on the concept for specifying a recording process. DVR MPEG-2 transport stream follows this model.

[0261]The input timing of MPEG-2 transport stream is explained. An input MPEG2 transport stream is a full transport stream or a partial transport stream. The MPEG2 transport stream inputted must follow ISO/IEC 13818-1 or ISO/IEC 13818-9. The i-th byte of an MPEG2 transport stream is simultaneously inputted into T-STD (it is prescribed by ISO/IEC 13818-1 Transport stream system target decoder) and saw spa KETTAIZA at time t(i). Rpk is the instant maximum of the input rate of a transport packet.

[0262]27MHz PLL52 generates the frequency of 27 MHz clocks. The frequency of 27 MHz clocks is locked by the value of PCR (Program Clock Reference) of MPEG-2 transport stream. arrival time clock counter53 is a binary counter which counts a pulse with a frequency of 27 MHz. Arrival_time_clock(i) is the counted value of Arrival time clock counter in time t(i).

[0263]source packetizer54 adds TP_extra_header to all the transport packets, and

makes a source packet. Arrival_time_stamp expresses the time when the 1st byte of a transport packet arrives to both T-STD and saw spa KETTAIZA. Arrival_time_stamp (k) is a sampled value of Arrival_time_clock (k), as shown in a following formula, and k shows the 1st byte of a transport packet here.

arrival_time_stamp(k) = arrival_time_clock(k) % 2^{30}[0264]When the time interval of two transport packets inputted continuously becomes 2^{30} / more than 27 million second (about 40 seconds), The difference of arrival_time_stamp of the two transport packets should be set as it has been 2^{30} / 27 million seconds. It prepares for the case where a recorder becomes such.

[0265]smoothing buffer55 carries out smoothing of the bit rate of an input transport stream. Don't overflow a smoothing buffer. Rmax is the output bit rate of the source packet from a smoothing buffer in case a smoothing buffer is not empty. When a smoothing buffer is empty, the output bit rate from a smoothing buffer is zero.

[0266]Next, the parameter of the recorder model of DVR MPEG-2 transport stream is explained. A value called Rmax is given by TS_recording_rate defined in ClipInfo() corresponding to an AV stream file. This value is computed by a following formula.

The value of Rmax = TS_recording_rate*192/188 TS_recording_rate is a size which makes bytes/second a unit.

[0267]When an input transport stream is a SESF transport stream, Rpk must be equal to TS_recording_rate defined in ClipInfo() corresponding to an AV stream file. When an input transport stream is not a SESF transport stream, Refer to the value set and defined without a descriptor, for example, maximum_bitrate_descriptor, partial_transport_stream_descriptor, etc. of MPEG-2 transport stream for this value.

[0268]When the input transport stream of smoothing buffer size is a SESF transport stream, the size of a smoothing buffer is zero. When an input transport stream is not a SESF transport stream, The size of a smoothing buffer The descriptor of MPEG-2 transport stream, For example, the value defined in smoothing_buffer_descriptor, short_smoothing_buffer_descriptor, partial_transport_stream_descriptor, etc. may be referred to.

[0269]A record machine (recorder) and the reproduction machine (player) must prepare the buffer of sufficient size. Default buffer size is 1536 bytes.

[0270]Next, the player model of DVR MPEG-2 transport stream is explained. Drawing 84 is a figure showing the player model of DVR MPEG-2 transport stream. This is a model on the concept for specifying reconstructive processing. DVR MPEG-2 transport stream follows this model.

[0271]X-tal61 generates 27 MHz of frequency of 27Mhz. The error span of 27-MHz

frequency must be +/−30 ppm (27 million+/− 810 Hz). arrival time clock counter62 is a binary counter which counts a pulse with a frequency of 27 MHz. Arrival_time_clock(i) is the counted value of Arrival time clock counter in time t(i).

[0272]In smoothing buffer64, Rmax is the input bit rate of the source packet to a smoothing buffer in case a smoothing buffer is not full. When a smoothing buffer is full, the input bit rate to a smoothing buffer is zero.

[0273]to explain the output timing of MPEG−2 transport stream, When arrival_time_stamp of the present source packet is equal to the value which is 30 bits of LSB of arrival_time_clock(i), the transport packet of the source packet is drawn out from a smoothing buffer. Rpk is the instant maximum of a transport packet rate. Don't carry out underflow of the smoothing buffer.

[0274]About the parameter of the player model of DVR MPEG−2 transport stream, it is the same as that of the parameter of the recorder model of DVR MPEG−2 transport stream mentioned above.

[0275]Drawing 85 is a figure showing the syntax of Source packet. transport_packet() is MPEG−2 transport packet specified by ISO/IEC 13818−1. The syntax of TP_Extra_header in the syntax of Source packet shown in drawing 85 is shown in drawing 86. It is an integer as which copy_permission_indicator expresses copy restrictions of the pay load of a transport packet for explaining the syntax of TP_Extra_header shown in drawing 86. Copy restrictions can be set to copy free, no more copy, copy once, or copy prohibited. Drawing 87 shows the value of copy_permission_indicator, and the relation in the mode specified by them.

[0276]copy_permission_indicator is added to all the transport packets. When recording an input transport stream using an IEEE1394 digital interface, the value of copy_permission_indicator, It may relate with the value of EMI (Encryption Mode Indicator) in IEEE1394 isochronouspacket header. The value of copy_permission_indicator may be related with the value of CCI embedded into the transport packet, when recording an input transport stream without using an IEEE1394 digital interface. The value of copy_permission_indicator may be related with the value of CGMS−A of an analog signal when carrying out self encoding of the analog signal input.

[0277]arrival_time_stamp is following formula arrival_time_stamp (k). In = arrival_time_clock(k) % 2^{30}, it is an integral value with the value specified by arrival_time_stamp.

[0278]The Clip AV stream must have [defining a Clip AV stream and] the structure of DVR MPEG−2 transport stream where a definition which was mentioned above is

carried out. arrival_time_clock(i) must increase continuously in a Clip AV stream. Even if the break point of system time base (STC base) exists in a Clip AV stream, arrival_time_clock(i) of the Clip AV stream must increase continuously.

[0279]The maximum of the start in a Clip AV stream and the difference of arrival_time_clock(i) between ends must be 26 hours. This restriction guarantees that PTS (Presentation Time Stamp) of the same value never appears in a Clip AV stream, when the break point of system time base (STC base) does not exist in an MPEG2 transport stream. The MPEG 2 systems standard has specified the wrap around cycle of PTS as 233 / 90000 second (about 26.5 hours).

[0280]The Bridge-Clip AV stream must have [defining a Bridge-Clip AV stream and] the structure of DVR MPEG-2 transport stream where a definition which was mentioned above is carried out. The Bridge-Clip AV stream must contain the break point of one arrival time base. The transport stream before and behind the break point of arrival time base must follow DVR-STD which must follow restriction of the coding mentioned later and is mentioned later.

[0281]In this embodiment, the video between PlayItem(s) in edit and seamless connection of an audio are supported. Making between PlayItem seamless connection guarantees "the continuous supply of data", and "seamless decoding processing" to a player/recorder. "The continuous supply of data" is being able to guarantee a file system supplying data by the required bit rate so that a decoder's may not be made to cause the underflow of a buffer. The real time nature of data is guaranteed, and data is stored by the block unit which sufficient size followed so that data can be read from a disk.

[0282]"Seamless decoding processing" is that a player can display the audio video data recorded on the disk, without making the reproducing output of a decoder start a pause and a gap.

[0283]The AV stream which PlayItem by which seamless connection is made refers to is explained. It can be judged whether connection of PlayItem to precede and the present PlayItem is guaranteed to indicate by seamless from the connection_condition field defined in the present PlayItem. The seamless connection between PlayItem(s) has the method of using Bridge-Clip, and a method which is not used.

[0284]Drawing 88 shows the relation between PlayItem preceded in the case of using Bridge-Clip, and the present PlayItem. In drawing 88, the stream data which a player reads give a shadow and are shown. TS1 shown in drawing 88 comprises the stream data which were able to attach the shadow of Clip1 (Clip AV stream), and the stream data which were able to attach the shadow before RSPN_arrival_time_discontinuity of

Bridge-Clip.

[0285]The stream data which were able to attach the shadow of Clip1 of TS1, From the address of a stream required in order to decode the presentation unit corresponding to IN_time (illustrated by IN_time1 in drawing 88) of PlayItem to precede, They are the stream data to the source packet referred to by RSPN_exit_from_previous_Clip. The stream data which were able to attach the shadow before RSPN_arrival_time_discontinuity of Bridge-Clip contained in TS1, They are the stream data from the source packet of the beginning of Bridge-Clip to the source packet in front of the source packet referred to by RSPN_arrival_time_discontinuity.

[0286]TS2 in drawing 88 comprises the stream data which were able to attach the shadow of Clip2 (Clip AV stream), and the stream data which were able to attach the shadow after RSPN_arrival_time_discontinuity of Bridge-Clip. The stream data which were able to attach the shadow after RSPN_arrival_time_discontinuity of Bridge-Clip contained in TS2, They are the stream data from the source packet referred to by RSPN_arrival_time_discontinuity to the source packet of the last of Bridge-Clip. The stream data which were able to attach the shadow of Clip2 of TS2, From the source packet referred to by RSPN_enter_to_current_Clip. They are the stream data to the address of a stream required in order to decode the presentation unit corresponding to OUT_time (illustrated by OUT_time2 in drawing 88) of the present PlayItem.

[0287]Drawing 89 shows the relation between PlayItem preceded when not using Bridge-Clip, and the present PlayItem. In this case, the stream data which a player reads give a shadow and are shown. TS1 in drawing 89 comprises the stream data which were able to attach the shadow of Clip1 (Clip AV stream). The stream data which were able to attach the shadow of Clip1 of TS1, It begins from the address of a stream required in order to decode the presentation unit corresponding to IN_time (illustrated by IN_time1 in drawing 89) of PlayItem to precede, and is data to the source packet of the last of Clip1. TS2 in drawing 89 comprises the stream data which were able to attach the shadow of Clip2 (Clip AV stream).

[0288]The stream data which were able to attach the shadow of Clip2 of TS2, They are the stream data to the address of a stream required in order to begin from the source packet of the beginning of Clip2 and to decode the presentation unit corresponding to OUT_time (illustrated by OUT_time2 in drawing 89) of the present PlayItem.

[0289]In drawing 88 and drawing 89, TS1 and T2 are the streams which the source packet followed. Next, stream regulation of TS1 and TS2 and the connection conditions between them are considered. First, the coding restrictions for seamless

connection are considered. As restriction of the coding structure of a transport stream, the number of the programs included in TS1 and TS2 must be 1 first. The number of the video streams contained in TS1 and TS2 must be 1. The number of the audio streams contained in TS1 and TS2 must be two or less. The number of the audio streams contained in TS1 and TS2 must be equal. In TS1 and/or TS2, elementary streams or private streams other than the above may be contained.

[0290]Restriction of a video bit stream is explained. Drawing 90 is a figure showing the example of the seamless connection shown by the display order of a picture. In order to be able to display a video stream seamlessly in a node, The unnecessary picture displayed the OUT_time1 (OUT_time of Clip1) back and before IN_time2 (IN_time of Clip2) must be removed by the process of re-encoding the partial stream of Clip near a node.

[0291]When shown in drawing 90, the example which makes seamless connection using BridgeSequence is shown in drawing 91. The video stream of Bridge-Clip before RSPN_arrival_time_discontinuity comprises the coding video stream to the picture corresponding to OUT_timeof Clip1 of drawing 90 1. And it is connected to the video stream of Clip1 to precede, and the video stream is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation.

[0292]Similarly, the video stream of Bridge-Clip after RSPN_arrival_time_discontinuity comprises the coding video stream after the picture corresponding to IN_timeof Clip2 of drawing 90 2. And a decoding start can be carried out correctly and it is connected to the video stream of Clip2 following this, and the video stream is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation. In order to make Bridge-Clip, generally, the picture of several sheets must be re-encoded and the other picture can be copied from original Clip.

[0293]The example which makes seamless connection without using BridgeSequence in the case of the example shown in drawing 90 is shown in drawing 92. The video stream of Clip1 comprises the coding video stream to the picture corresponding to OUT_time1 of drawing 90, and it is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation. Similarly, the video stream of Clip2 comprises the coding video stream after the picture corresponding to IN_timeof Clip2 of drawing 90 2, and it is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation.

[0294]The frame rate of the video stream of TS1 and TS2 must be equal to explaining coding restrictions of a video stream first. The termination of the video stream of TS1

must be carried out by sequence_end_code. The video stream of TS2 must be started by Sequence Header, GOP Header, and I-picture. The video stream of TS2 must be started by closed GOP.

[0295]The video presentation unit (a frame or the field) defined in a bit stream must be continuation on both sides of a node. There must not be any gap of a frame or the field in a node. In a node, the field sequence of a top ? bottom product must be continuation. In encoding which uses 3-2 PURUDAUN, "top_field_first" It reaches. In order to rewrite a "repeat_first_field" flag or to prevent generating of a field gap, it may be made to re-encode locally.

[0296]If the sampling frequency of the audio of TS1 and TS2 is not the same, it will not be explaining coding restrictions of an audio bit stream. If the encoding method (example . the MPEG1 layer 2, AC-3, SESF LPCM, AAC) of the audio of TS1 and TS2 is not the same, it will not become.

[0297]Next, the audio frame of the last of the audio stream of TS1 must contain the audio sample with display time equal at the time of the end of a display of the display picture of the last of TS1 in explaining coding restrictions of MPEG-2 transport stream. The audio frame of the beginning of the audio stream of TS2 must contain the audio sample with display time equal at the time of the display start of the display picture of the beginning of TS2.

[0298]In a node, the sequence of an audio presentation unit must not have a gap. As shown in drawing 93, there may be overlap defined by the length of the audio presentation unit of less than 2 audio frame sections. The first packet that transmits the elementary stream of TS2 must be a video packet. The transport stream in a node must follow DVR-STD mentioned later.

[0299]TS1 and TS2 must not contain the break point of arrival time base in explaining restriction of Clip and Bridge-Clip in each.

[0300]The following restrictions are applied only when using Bridge-Clip. Only in the node of the source packet of the last of TS1, and the source packet of the beginning of TS2, a Bridge-ClipAV stream has a break point of only one arrival time base. RSPN_arrival_time_discontinuity defined in ClipInfo() must show the address of the break point, and it must show the address which refers to the source packet of the beginning of TS2.

[0301]May any source packet in Clip1 be sufficient as the source packet referred to by RSPN_exit_from_previous_Clip defined in BridgeSequenceInfo()? It does not need to be a boundary of Aligned unit. May any source packet in Clip2 be sufficient as the source packet referred to by RSPN_enter_to_current_Clip defined in

BridgeSequenceInfo()? It does not need to be a boundary of Aligned unit.

[0302]OUT_time (OUT_time1 shown in drawing 88 and drawing 89) of PlayItem preceded for explaining restriction of PlayItem must show the display finish time of the video presentation unit of the last of TS1. IN_time (IN_time2 shown in drawing 88 and drawing 89) of the present PlayItem must show the display start time of the video presentation unit of the beginning of TS2.

[0303]Seamless connection must be made by explaining restriction of the data allocation in the case of using Bridge-Clip with reference to drawing 94 so that the continuous supply of data may be guaranteed by a file system. This must be performed by arranging the Bridge-Clip AV stream connected to Clip1 (Clip AV stream file) and Clip2 (Clip AV stream file) so that data allocation regulation may be fulfilled.

[0304]The stream portion of Clip1 (Clip AV stream file) before RSPN_exit_from_previous_Clip as arranged to the continuation field more than half fragmentation, RSPN_exit_from_previous_Clip must be chosen. The data length of a Bridge-Clip AV stream must be chosen so that it may be arranged to the continuation field more than half fragmentation. The stream portion of Clip2 (Clip AV stream file) after RSPN_enter_to_current_Clip as arranged to the continuation field more than half fragmentation, RSPN_enter_to_current_Clip must be chosen.

[0305]Seamless connection must be made by explaining restriction of the data allocation in the case of making seamless connection without using Bridge-Clip with reference to drawing 95 so that the continuous supply of data may be guaranteed by a file system. This must be performed by arranging the portion of the last of Clip1 (Clip AV stream file), and the portion of the beginning of Clip2 (Clip AV stream file) so that data allocation regulation may be fulfilled.

[0306]The stream portion of the last of Clip1 (Clip AV stream file) must be arranged to the continuation field more than half fragmentation. The stream portion of the beginning of Clip2 (Clip AV stream file) must be arranged to the continuation field more than half fragmentation.

[0307]When the digital AV signal with the predetermined bit rate is fragmented and recorded on the disk, In order to guarantee that the recorded digital AV signal can be read from the recording medium 100 by the predetermined bit rate, the size of one continuous recording field must fulfill the following conditions.

S*8 / (S*8/Rud+Ts) >= Rmax -- here, S : The minimum size of one continuous recording field . [Byte] -- access time [second] Rud: of the full strokes from a Ts:1 ** record section to the next record section -- the read-out bit rate from an archive

medium [bit/second] Rmax: the bit rate [bit/second] of an AV stream -- that is, Data must be arranged so that the data of an AV stream may continue and may be recorded on a disk more than S byte.

[0308]The size of the above-mentioned half fragmentation must arrange data so that it may become more than S byte.

[0309]Next, DVR-STD is explained. DVR-STD is a conceptual model for modeling generation of a DVR MPEG2 transport stream, and decoding in the case of verification. DVR-STD is also a conceptual model for modeling generation of the AV stream referred to by two PlayItem(s) which were mentioned above, and by which seamless connection was made, and decoding in the case of verification.

[0310]A DVR-STD model is shown in drawing 96. The DVR MPEG-2 transport-stream player model is contained in the model shown in drawing 96 as a component. The transcription method of n, TBn, MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, On, and Pn (k) is the same as what is defined as T-STD of ISO/IEC 13818-1. That is, it is as follows. n is an index number of an elementary stream. TBn is a transport buffer of the elementary stream n, and is **.

[0311]MBn is a multiple buffer of the elementary stream n. It exists only about a video stream. EBn is an elementary stream buffer of the elementary stream n. It exists only about a video stream. TBsys is an input buffer for the system information of the program under decoding. Bsys is a main buffer in the system target decoder for the system information of the program under decoding. Rxn is a transmission rate by which data is removed from TBn. Rbxn is a transmission rate by which a PES packet pay load is removed from MBn. It exists only about a video stream.

[0312]Rxsys is a transmission rate by which data is removed from TBsys. Dn is a decoder of the elementary stream n. Dsys is a decoder about the system information of the program under decoding. On is re-ordering buffer of the video stream n. Pn (k) is the k-th presentation unit of the elementary stream n.

[0313]The decoding process of DVR-STD is explained. While reproducing DVR MPEG-2 single transport stream, the timing which inputs a transport packet into the buffer of TB1, TBn, or TBsys is determined by arrival_time_stamp of a source packet. Regulation of the buffering operation of TB1, MB1, EB1, TBn, Bn, TBsys, and Bsys is the same as T-STD specified to ISO/IEC 13818-1. Regulation of decoding operation and a display action is the same as T-STD specified to ISO/IEC 13818-1.

[0314]The decoding process [it is reproducing PlayItem by which seamless connection was made] of a between is explained. Here, reproduction of two AV streams referred to by PlayItem by which seamless connection was made will be

explained, and future explanation explains the reproduction of TS (for example, shown in drawing 88)1, and TS2 mentioned above. TS1 is a stream to precede and TS2 is the present stream.

[0315]Drawing 97 shows the timing chart of the input of a transport packet when moving from a certain AV stream (TS1) to the following AV stream (TS2) seamlessly connected to it, decoding, and a display. While moving from a predetermined AV stream (TS1) to the following AV stream (TS2) seamlessly connected to it, The time-axis (drawing 97 is shown by ATC2) of the arrival time base of TS2 is not the same as that (drawing 97 is shown by ATC1) of the arrival time base of TS1.

[0316]The time-axis (drawing 97 is shown by STC2) of the system time base of TS2 is not the same as that (drawing 97 is shown by STC1) of the system time base of TS1. It is required that the display of video should continue seamlessly. Overlap may be shown in the display time of the presentation unit of an audio.

[0317]The input timing to DVR-STD is explained. Until the video packet of the time to time T_1, i.e., the last of TS1, carries out the end of an input TB1 of DVR-STD, The input timing to the buffer of TB1 of DVR-STD, TBn, or TBsys is determined by arrival_time_stamp of the source packet of TS1.

[0318]The remaining packets of TS1 must be inputted into the buffer of TBn of DVR-STD, or TBsys by the bit rate of TS_recording_rate (TS1). Here, TS_recording_rate (TS1) is a value of TS_recording_rate defined in ClipInfo() corresponding to Clip1. The time which the byte of the last of TS1 inputs into a buffer is time T_2. Therefore, arrival_time_stamp of a source packet is disregarded in the section from time T_1 to T_2.

[0319]When N1 is made into the number of bytes of the transport packet of TS1 following the video packet of the last of TS1, time DT1 to time T_1 thru/or T_2, It is time required in order that N1 byte may carry out the end of an input by the bit rate of TS_recording_rate (TS1), and is computed by a following formula.

deltaT1=T_2-T_1=N1 / TS_recording_rate Both the values of RXn and RXsys change to the value of TS_recording_rate (TS1) before time (TS1) T_1 thru/or T_2. Buffering operation other than this rule is the same as T-STD.

[0320]arrival time clock counter is reset by the value of arrival_time_stamp of the source packet of the beginning of TS2 in the time of T_2. The input timing to the buffer of TB1 of DVR-STD, TBn, or TBsys is determined by arrival_time_stamp of the source packet of TS2. RXn and RXsys both change to the value defined in T-STD.

[0321]About additional audio buffering and system-data buffering to explain an audio decoder and a system decoder, In addition to the amount of buffers defined by T-STD,

the additional amount of buffers (data volume for about 1 second) is required so that the input data of the section from the time T1 to T2 can be processed.

[0322]The display of a video presentation unit must let a node pass to explain the presentation timing of video, and it must be continuation without a gap. Here, STC1 considers it as the time-axis (in drawing 97, illustrated with STC1) of the system time base of TS1, and STC2 is a time-axis (in drawing 97, illustrated with STC2.) of the system time base of TS2. Correctly, STC2 is started from the time which PCR of the beginning of TS2 inputted into T-STD. It carries out.

[0323]The offset between STC1 and STC2 is determined as follows. PTS^1_{end} is PTS on STC1 corresponding to the video presentation unit of the last of TS1, and PTS^2_{start}, Are PTS on STC2 corresponding to the video presentation unit of the beginning of TS2, and T_{pp}, If it is a display period of the video presentation unit of the last of TS1, offset STC_delta between two system time base will be computed by a following formula.

$STC\_delta = PTS^1_{end} + T_{pp} - PTS^2_{start}$[0324]In a node to explain the timing of the presentation of an audio, There may be overlap of the display timing of an audio presentation unit, and it is 0 thru/or less than 2 audio frames (see "audio overlap" currently illustrated by drawing 97). Which audio sample being chosen and carrying out resynchronization of the display of an audio presentation unit to the amended time base after a node are set up by the player side.

[0325]In time T_5, the audio presentation unit of the last of TS1 is displayed for explaining about the system time clock of DVR-STD. The system time clock may overlap from time T_2 to T_5. In this section, DVR-STD changes a system time clock between the value (STC1) of old time base, and the value (STC2) of new time base. The value of STC2 is computed by a following formula.

STC2=STC1-STC_delta[0326]The continuity of buffering is explained. $STC1^1_{video\_end}$ is a value of STC on system time base STC1 in case the byte of the last of the video packet of the last of TS1 arrives to TB1 of DVR-STD. $STC2^2_{video\_start}$ is a value of STC on system time base STC2 in case the byte of the beginning of the video packet of the beginning of TS2 arrives to TB1 of DVR-STD. $STC2^1_{video\_end}$ is the value which converted the value of $STC1^1_{video\_end}$ into the value on system time base STC2. $STC2^1_{video\_end}$ is computed by a following formula.

$STC2^1_{video\_end} = STC1^1_{video\_end} - STC\_delta$ [0327]In order to follow DVR-STD, it is required that the following two conditions should be fulfilled. First, the arrival timing of TB1 of the video packet of the beginning of TS2 must fill the inequality shown below. And the inequality shown below must be filled.

It reaches Clip1 so that the inequality of $STC2^2_{video\_start} > STC2^1_{video\_end} + deltaT_1$ ** may be filled, or the partial stream of Clip2 -- re-encoding -- and -- or when it is necessary to re-multiplex-ize, it is carried out if needed [the].

[0328]next, the input of the video packet from TS2 which continues at the input of the video packet from TS1, and it on the time-axis of the system time base which converted STC1 and STC2 on the same time-axis -- a video buffer -- overflow -- and don't carry out underflow.

[0329]Drawing 98 is a figure showing example of another of the syntax of BridgeSequenceInfo(). The difference from BridgeSequenceInfo() of drawing 38 is that only Bridge_Clip_Information_file_name is contained.

[0330]Drawing 99 is a figure with which two PlayItem(s) explain Bridge-Clip when connected seamlessly, when using the syntax of BridgeSequenceInfo() of drawing 98. RSPN_exit_from_previous_Clip, It is a source packet number of the source packet on Clip AVstream which PlayItem to precede refers to, and the source packet of the beginning of a Bridge-Clip AV stream file is connected following this source packet.

[0331]RSPN_enter_to_current_Clip, It is a number of the source packet on Clip AV stream which the present PlayItem refers to, and the source packet of the last of a Bridge-Clip AV stream file is connected before this source packet. In the Bridge-Clip AV stream file shown in drawing 99, SPN_ATC_start shows the source packet number of the source packet which the time-axis of new arrival time base starts in a Bridge-Clip AV stream file.

[0332]A Bridge-Clip AV stream file has a break point of one arrival time base. 2nd SPN_ATC_start has the same meaning as RSPN_arrival_time_discontinuity of drawing 37 in a figure.

[0333]When the syntax of BridgeSequenceInfo() of drawing 98 is used, RSPN_exit_from_previous_Clip and RSPN_enter_to_current_Clip are stored into the Clip Information file corresponding to a Bridge-Clip AV stream file. SPN_ATC_start is also stored into a Clip Information file.

[0334]Drawing 100 is a figure in which BridgeSequenceInfo shows the syntax of the ClipInformation file in the case of the syntax of drawing 98. SequenceInfo_start_address shows the start address of SequenceInfo() by making the relative number of bytes from the byte of the head of a Clip Information file into a unit. A relative number of bytes is counted from zero.

[0335]Drawing 101 is a figure showing the syntax of ClipInfo() of the Clip Information file of drawing 100. Clip_stream_type shows whether the AV stream file of the Clip is a ClipAV stream file, or it is a Bridge-Clip AV stream file. When Clip_stream_type shows

a Bridge-Clip AV stream file, the next syntax field continues.

[0336]previous_Clip_Information_file_name shows the Clip Information file name of Clip connected before the Bridge-Clip AV stream file. RSPN_exit_from_previous_Clip, It is a source packet number of the source packet on the ClipAV stream file shown by previous_Clip_Information_file_name, The source packet of the beginning of a Bridge-Clip AV stream file is connected following the source packet. The source packet number is a value which counts zero as an initial value from the source packet of the beginning of a Clip AV stream file.

[0337]current_Clip_Information_file_name shows the Clip Information file name of Clip connected behind the Bridge-Clip AV stream file. RSPN_enter_to_current_Clip, It is a source packet number of the source packet on the Clip AV stream file shown by current_Clip_Information_file_name, The source packet of the last of a Bridge-Clip AV stream file is connected before the source packet. The source packet number is a value which counts zero as an initial value from the source packet of the beginning of a Clip AV stream file.

[0338]Drawing 102 shows the syntax of SequenceInfo() of the Clip Information file of drawing 100. num_of_ATC_sequences shows the number of ATC-sequence in an AV stream file. ATC-sequence is a source packet sequence which does not contain the break point of arrival time base. In Bridge-Clip, this value is 2.

[0339]SPN_ATC_start [atc_id] The address which the arrival time base to which it is pointed out by atc_id on an AV stream file starts is shown. SPN_ATC_start [atc_id] is a size which makes a source packet number a unit, and counts zero as an initial value from the source packet of the beginning of an AV stream file.

[0340]Clip by which Drawing 103 is referred to by Bridge-Sequence It is a figure explaining change of the database at the time of eliminating the stream data of an AV stream file selectively. "Before Editing" of Drawing 103 (A) -- Shimesu -- like, it is connected by Bridge-Clip and Clip1 and Clip2, Suppose that they are RSPN_exit_from_previous_Clip=X and RSPN_exit_from_previous_Clip=Y.

[0341]Suppose that the stream-data portion of the Z1 piece source packet shown with the slash of Clip1 and the stream-data portion of the Z2 piece source packet shown with the slash of Clip2 are eliminated at this time. As a result, a value is changed into RSPN_exit_from_previous_Clip=X-Z1 and RSPN_exit_from_previous_Clip=Y-Z2 as "After Editing" of Drawing 103 (B) shows.

[0342]By changing the syntax of the database which is related to BridgeSequence, as shown in drawing 98 and Drawing 101, Information about a source packet number which shows the data address in an AV stream (in the syntax of a database) The field

which starts with RSPN will disappear from the layer of PlayList, and all the information on a source packet number will be described by the layer of Clip.

[0343]When change is needed for the value of the data address in an AV stream by this (for example, this is needed when the data of an AV stream file is eliminated selectively), Since what is necessary is just to carry out data management only of the Clip information files, there is a merit to which management of a database becomes easy.

[0344]Drawing 104 is a flow chart explaining creation of Real PlayList. It explains referring to the block diagram of the recording and reproducing device 1 of drawing 1. In Step S10, the control section 23 records a Clip AV stream. In Step S11, the control section 23 creates PlayList() which consists of PlayItem which covers all the refreshable ranges of the above-mentioned Clip. An STC break point is in Clip, and when PlayList() consists of two or more PlayItem(s), connection_condition between PlayItem(s) is also determined.

[0345]In Step S12, the control section 23 creates UIAppInfoPlayList(). In Step S13, the control section 23 creates PlayListMark. In Step 14, the control section 23 creates MakersPrivateData. In Step S15, the control section 23 records a Real PlayList file. Thus, whenever it records a Clip AV stream newly, one Real PlayList file is made.

[0346]Drawing 105 is a flow chart explaining the creation with a bridge sequence of Virtual PlayList. In Step S20, it lets a user interface pass and playback of one Real PlayList currently recorded on the disk is specified. And out of the reproduction range of the Real PlayList, it lets a user interface pass and the reproducing section shown by the IN point and an OUT point is specified.

[0347]When it is judged that it progresses to step SS22 in Step S21 when it is judged that the control section 23 judged whether all the designating operation of the reproduction range by a user was completed, and is completed, and it has not ended, it returns to Step S20 and processing after it is repeated.

[0348]In Step S22, a user determines the connected state (connection_condition) between two PlayItem(s) reproduced continuously through a user interface, or the control section 23 is determined. In Step S23, the control section 23 creates the bridge sequence for PlayItem by which seamless connection is made. In Step S24, the control section 23 creates and records a Virtual PlayList file.

[0349]Drawing 106 is a flow chart explaining the detailed processing in Step S23. In Step S31, the control section 23 performs re-encoding and re-multiplex-izing of the AV stream by the side of the OUT point of PlayItem displayed on a front side in time.

In Step S32, the control section 23 performs re-encoding and re-multiplex-izing of the AV stream by the side of the IN point of PlayItem displayed following the above-mentioned PlayItem.

[0350]In Step S33, the control section 23 determines the value of RSPN_exit_from_previous_Clip so that the data allocation conditions for the continuous supply of data may be fulfilled. Namely, as the stream portion of the Clip AV stream file before RSPN_exit_from_previous_Clip is arranged to the continuation field more than the above-mentioned half fragmentation on the recording medium, RSPN_exit_from_previous_Clip must be chosen (see drawing 91 and drawing 94).

[0351]In Step S34, the control section 23 determines the value of RSPN_enter_to_current_Clip so that the data allocation conditions for the continuous supply of data may be fulfilled. Namely, as the stream portion of the Clip AV stream file after RSPN_enter_to_current_Clip is arranged to the continuation field more than the above-mentioned half fragmentation on the recording medium 100, RSPN_enter_to_current_Clip must be chosen (see drawing 91 and drawing 94).

[0352]In Step S35, the control section 23 creates a Bridge-Clip AV stream file so that the data allocation conditions for the continuous supply of data may be fulfilled. That is, when the quantity of the data created by processing of Step S31 and Step S32 is less than the size more than the above-mentioned half fragmentation, data is copied from original Clip and Bridge-Clip is created (see drawing 91 and drawing 94).

[0353]Although each processing of Step S33, S34, and S35 is explained to a time series, since each is [these processings] related, processing may be performed in random order or simultaneous.

[0354]In Step S36, the control section 23 creates the database of a bridge sequence. In Step S37, the control section 23 records a Bridge-Clip AV stream file and its Clip information files. Thus, out of the playback range of Real PlayList currently recorded on the disk. One or more PlayItem(s) are chosen by the user, the bridge sequence for making seamless connection of between two PlayItem(s) is created, and one or more PlayItem(s) are recorded considering that by which grouping was carried out as one Virtual PlayList file.

[0355]Drawing 107 is a flow chart explaining reproduction of PlayList. In Step S41, the control section 23 acquires the information on Info.dvr, Clip Information file, PlayList file, and a thumbnail file, The GUI picture in which the list of PlayList currently recorded on the disk is shown is created, and it lets a user interface pass, and displays on GUI.

[0356]In Step S42, the control section 23 shows a GUI picture the information

explaining PlayList based on UIAppInfoPlayList() of each PlayList. In Step S43, it lets a user interface pass and a user directs reproduction of one PlayList from on a GUI picture. In Step S44, the control section 23 acquires the source packet number which has the nearest entry point in front in time than IN_time from STC-sequence-id of the present PlayItem, and PTS of IN_time.

[0357]In Step S45, the control section 23 reads the data of an AV stream from a source packet number with the above-mentioned entry point, and supplies it to a decoder. In Step S46, when there is front PlayItem in time [the present PlayItem], the control section 23 performs connection processing of the display with front PlayItem and the present PlayItem according to connection_condition. When seamless connection of the PlayItem is made, an AV stream is decoded based on the decoding method of DVR-STD.

[0358]In Step S47, the control section 23 directs to start a display from the picture of PTS of IN_time to AV decoder 27. In Step S48, the control section 23 directs to continue decoding of an AV stream to AV decoder 27. In Step S49, the control section 23 judges whether the picture of the present display is a picture of PTS of OUT_time, When it is judged that it is not a picture of PTS of OUT_time, after progressing to Step S50 and displaying a picture, it returns to Step S48 and processing after it is repeated.

[0359]On the other hand, in Step S49, when the picture of the present display is judged to be a picture of PTS of OUT_time, it progresses to Step S51. In Step S51, as for the control section 23, the present PlayItem judges in PlayList whether it is the last PlayItem, When it is judged that it is not the last PlayItem, it returns to Step S44 and processing after it is repeated, and reproduction of PlayList is ended when it is judged that it is the last PlayItem.

[0360]Thus, reproduction of one PlayList file in which reproduction instruction was done by the user is performed.

[0361]The contents of the data currently recorded on the recording medium by being based on such syntax, a data structure, and a rule, Reproduction information etc. can be managed appropriately and it has them, and a user can check the contents of the data currently appropriately recorded on the recording medium at the time of reproduction, or it can make it possible to reproduce desired data simple.

[0362]Although a series of processings mentioned above can also be performed by hardware, they can also be performed with software. The computer by which the program which constitutes the software is included in hardware for exclusive use when performing a series of processings with software, Or it is installed in the personal computer etc. which can perform various kinds of functions, for example, are

general-purpose, etc. from a recording medium by installing various kinds of programs.
[0363]Drawing 108 is a figure showing the example of an internal configuration of a general-purpose personal computer. CPU(Central Processing Unit) 201 of a personal computer performs various kinds of processings according to the program memorized by ROM(ReadOnly Memory) 202. In RAM(Random Access Memory) 203, CPU201 performs various kinds of processings, and also required data, a program, etc. are suitably memorized. The input part 206 which comprises a keyboard and a mouse is connected, and the input/output interface 205 outputs the signal inputted into the input part 206 to CPU201. The outputting part 207 which comprises a display, a loudspeaker, etc. is also connected to the input/output interface 205.
[0364]The communications department 209 which performs transfer of other devices and data via networks, such as the storage parts store 208 which comprises a hard disk etc., and the Internet, is also connected to the input/output interface 205. The drive 210 is used, when reading data from recording media, such as the magnetic disk 221, the optical disc 222, the magneto-optical disc 223, and the semiconductor memory 224, or writing in data.
[0365]. As shown in Drawing 108, this recording medium is distributed apart from a computer in order to provide a user with a program. The magnetic disk 221 (a floppy disk is included) with which the program is recorded, the optical disc 222 (CD-ROM (Compact Disk-Read Only Memory).) . DVD (Digital Versatile Disk) is included. It is not only constituted by the package media which consist of the magneto-optical disc 223 (MD (Mini-Disk) is included) or the semiconductor memory 224, but, It comprises a hard disk etc. in which ROM202 with which a user is provided in the state where it was beforehand included in the computer, and the program is remembered to be, and the storage parts store 208 are contained.
[0366]In this specification, even if the processing serially performed according to an order that the step which describes the program provided by a medium was indicated is not of course necessarily processed serially, it also includes a parallel target or the processing performed individually.
[0367]In this specification, a system expresses the whole device constituted by two or more devices.
[0368]
[Effect of the Invention]According to the information processor of this invention, a method, and the program, like the above. When being continuously reproduced from the 1st AV stream to the 2nd AV stream is directed, While generating the 3rd AV stream reproduced when it comprises a predetermined portion of the 1st AV stream,

and a predetermined portion of the 2nd AV stream and reproduction is switched to the 2nd AV stream from the 1st AV stream, The information on the address of the source packet of the 1st AV stream in the timing which changes reproduction from the 1st AV stream to the 3rd AV stream as information relevant to the 3rd AV stream, Since the address information which comprises information on the address of the source packet of the 2nd AV stream in the timing which changes reproduction from the 3rd AV stream to the 2nd AV stream was generated, it is renewable so that the continuity of the AV stream recorded independently may be maintained.

[0369]According to the 2nd information processor of this invention, a method, and the program, the 1st AV stream, the 2nd AV stream, Or read the 3rd AV stream from a recording medium, and as information relevant to the 3rd AV stream, The information on the address of the source packet of the 1st AV stream in the timing which changes reproduction from the 1st AV stream to the 3rd AV stream, The address information which comprises information on the address of the source packet of the 2nd AV stream in the timing which changes reproduction from the 3rd AV stream to the 2nd AV stream is read from a recording medium, Based on the information relevant to the 3rd read AV stream, reproduction is changed from the 1st AV stream to the 3rd AV stream, Since reproduction is changed from the 3rd AV stream to the 2nd AV stream and it was made to reproduce, it is renewable so that the continuity of the AV stream recorded independently may be maintained.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a figure showing the composition of the 1 embodiment of the recording and reproducing device which applied this invention.

[Drawing 2]It is a figure explaining the format of the data recorded on a recording medium by the recording and reproducing device 1.

[Drawing 3]It is a figure explaining Real PlayList and Virtual PlayList.

[Drawing 4]It is a figure explaining creation of Real PlayList.

[Drawing 5]It is a figure explaining deletion of Real PlayList.

[Drawing 6]It is a figure explaining assemble editing.

[Drawing 7]It is a figure explaining the case where a sub path is provided in Virtual PlayList.

[Drawing 8]It is a figure explaining change of the reproduction sequence of PlayList.

[Drawing 9]It is a figure explaining the mark on PlayList, and the mark on Clip.

[Drawing 10]It is a figure explaining a menu thumbnail.

[Drawing 11]It is a figure explaining the mark added to PlayList.

[Drawing 12]It is a figure explaining the mark added to a clip.

[Drawing 13]It is a figure explaining the relation of PlayList, Clip, and a thumbnail file.

[Drawing 14]It is a figure explaining directory structure.

[Drawing 15]It is a figure showing the syntax of info.dvr.

[Drawing 16]It is a figure showing the syntax of DVR volume.

[Drawing 17]It is a figure showing the syntax of Resumevolume.

[Drawing 18]It is a figure showing the syntax of UIAppInfovolume.

[Drawing 19]It is a figure showing the table of Character set value.

[Drawing 20]It is a figure showing the syntax of TableOfPlayList.

[Drawing 21]It is a figure showing other syntax of TableOfPlayList.

[Drawing 22]It is a figure showing the syntax of MakersPrivateData.

[Drawing 23]xxxxx. It is a figure showing the syntax of rpls and yyyyy.vpls.

[Drawing 24]It is a figure explaining PlayList.

[Drawing 25]It is a figure showing the syntax of PlayList.

[Drawing 26]It is a figure showing the table of PlayList_type.

[Drawing 27]It is a figure showing the syntax of UIAppinfoPlayList.

[Drawing 28]It is a figure explaining the flag in the syntax of UIAppinfoPlayList shown in drawing 27.

[Drawing 29]It is a figure explaining PlayItem.
[Drawing 30]It is a figure explaining PlayItem.
[Drawing 31]It is a figure explaining PlayItem.
[Drawing 32]It is a figure showing the syntax of PlayItem.
[Drawing 33]It is a figure explaining IN_time.
[Drawing 34]It is a figure explaining OUT_time.
[Drawing 35]It is a figure showing the table of Connection_Condition.
[Drawing 36]It is a figure explaining Connection_Condition.
[Drawing 37]It is a figure explaining BridgeSequenceInfo.
[Drawing 38]It is a figure showing the syntax of BridgeSequenceInfo.
[Drawing 39]It is a figure explaining SubPlayItem.
[Drawing 40]It is a figure showing the syntax of SubPlayItem.
[Drawing 41]It is a figure showing the table of SubPath_type.
[Drawing 42]It is a figure showing the syntax of PlayListMark.
[Drawing 43]It is a figure showing the table of Mark_type.
[Drawing 44]It is a figure explaining Mark_time_stamp.
[Drawing 45]It is a figure showing the syntax of zzzzz.clip.
[Drawing 46]It is a figure showing the syntax of ClipInfo.
[Drawing 47]It is a figure showing the table of Clip_stream_type.
[Drawing 48]It is a figure explaining offset_SPN.
[Drawing 49]It is a figure explaining offset_SPN.
[Drawing 50]It is a figure explaining the STC section.
[Drawing 51]It is a figure explaining STC_Info.
[Drawing 52]It is a figure showing the syntax of STC_Info.
[Drawing 53]It is a figure explaining ProgramInfo.
[Drawing 54]It is a figure showing the syntax of ProgramInfo.
[Drawing 55]It is a figure showing the syntax of VideoCondingInfo.
[Drawing 56]It is a figure showing the table of Video_format.
[Drawing 57]It is a figure showing the table of frame_rate.
[Drawing 58]It is a figure showing the table of display_aspect_ratio.
[Drawing 59]It is a figure showing the syntax of AudioCondingInfo.
[Drawing 60]It is a figure showing the table of audio_coding.
[Drawing 61]It is a figure showing the table of audio_component_type.
[Drawing 62]It is a figure showing the table of sampling_frequency.
[Drawing 63]It is a figure explaining CPI.
[Drawing 64]It is a figure explaining CPI.

[Drawing 65]It is a figure showing the syntax of CPI.
[Drawing 66]It is a figure showing the table of CPI_type.
[Drawing 67]It is a figure explaining video EP_map.
[Drawing 68]It is a figure explaining EP_map.
[Drawing 69]It is a figure explaining EP_map.
[Drawing 70]It is a figure showing the syntax of EP_map.
[Drawing 71]It is a figure showing the table of EP_type values.
[Drawing 72]It is a figure showing the syntax of EP_map_for_one_stream_PID.
[Drawing 73]It is a figure explaining TU_map.
[Drawing 74]It is a figure showing the syntax of TU_map.
[Drawing 75]It is a figure showing the syntax of ClipMark.
[Drawing 76]It is a figure showing the table of mark_type.
[Drawing 77]It is a figure showing the table of mark_type_stamp.
[Drawing 78]It is a figure showing the syntax of menu.thmb and mark.thmb.
[Drawing 79]It is a figure showing the syntax of Thumbnail.
[Drawing 80]It is a figure showing the table of thumbnail_picture_format.
[Drawing 81]It is a figure explaining tn_block.
[Drawing 82]It is a figure explaining the structure of the transport stream of DVR MPEG 2.
[Drawing 83]It is a figure showing the recorder model of the transport stream of DVR MPEG 2.
[Drawing 84]It is a figure showing the player model of the transport stream of DVR MPEG 2.
[Drawing 85]It is a figure showing the syntax of source packet.
[Drawing 86]It is a figure showing the syntax of TP_extra_header.
[Drawing 87]It is a figure showing the table of copy permission indicator.
[Drawing 88]It is a figure explaining seamless connection.
[Drawing 89]It is a figure explaining seamless connection.
[Drawing 90]It is a figure explaining seamless connection.
[Drawing 91]It is a figure explaining seamless connection.
[Drawing 92]It is a figure explaining seamless connection.
[Drawing 93]It is a figure explaining the overlap of an audio.
[Drawing 94]It is a figure explaining the seamless connection using BridgeSequence.
[Drawing 95]It is a figure explaining the seamless connection which does not use BridgeSequence.
[Drawing 96]It is a figure showing a DVR STD model.

[Drawing 97]It is a figure showing the timing chart of decoding and a display.
[Drawing 98]It is a figure showing other syntax of BridgeSequenceInfo.
[Drawing 99]It is a figure explaining Bridge-Clip in case two PlayItem(s) are connected seamlessly.
[Drawing 100]It is a figure showing the syntax of a ClipInformation file.
[Drawing 101]It is a figure showing the syntax of ClipInfo of a ClipInformation file.
[Drawing 102]It is a figure showing the syntax of SequenceInfo of a ClipInformation file.
[Drawing 103]It is a figure explaining change of the database at the time of eliminating the stream data of a ClipAV stream file selectively.
[Drawing 104]It is a flow chart explaining creation of RealPlayList.
[Drawing 105]It is a flow chart explaining creation of VirtualPlayList.
[Drawing 106]It is a flow chart explaining creation of a bridge sequence.
[Drawing 107]It is a flow chart explaining reproduction of PlayList.
[Drawing 108]It is a figure explaining a medium.
[Description of Notations]
1 A recording and reproducing device, and 11 thru/or 13 A terminal, 14 Analyzing parts and 15 AV encoder, 16 A multiplexer and 17 A switch, 18 Multiplexed stream analyzing parts and 19 [A user interface and 25 / A switch and 26 / A demultiplexer,] Saw spa KETTAIZA, 20 ECC-code-ized part, and 21 A modulation part, 22 writing parts, and 23 A control section and 24 27 An AV decoder, 28 read sections, and 29 A demodulation section, 30 ECC decoding part, and 31 Saw spa KETTAIZA, 32, and 33 Terminal

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2002－158974
（P2002－158974A）
(43)公開日　平成14年５月31日(2002.5.31)

(51)Int.Cl.⁷		識別記号	FI		テーマコード(参考)	
Ｈ０４Ｎ	5/93		Ｇ１１Ｂ	20/10	３２１Ｚ	５Ｃ０５２
Ｇ１１Ｂ	20/10	３２１		20/12		５Ｃ０５３
	20/12				１０３	５Ｃ０５９
		１０３	Ｈ０４Ｎ	5/85	Ａ	５Ｄ０４４
Ｈ０４Ｎ	5/85			5/93	Ｚ	

審査請求　未請求　請求項の数14　ＯＬ　（全 63 頁）　最終頁に続く

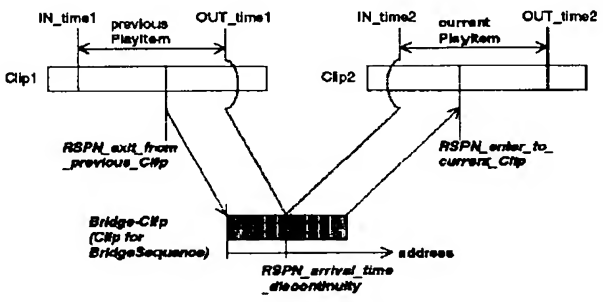
(21)出願番号　特願2001－109341(P2001－109341)
(22)出願日　平成13年４月６日(2001.4.6)
(31)優先権主張番号　特願2000－183769(P2000－183769)
(32)優先日　平成12年４月21日(2000.4.21)
(33)優先権主張国　日本（ＪＰ）
(31)優先権主張番号　特願2000－271550(P2000－271550)
(32)優先日　平成12年９月７日(2000.9.7)
(33)優先権主張国　日本（ＪＰ）

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　加藤　元樹
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(72)発明者　浜田　俊也
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　100082131
弁理士　稲本　義雄

最終頁に続く

(54)【発明の名称】　情報処理装置および方法、記録媒体、プログラム、並びに記録媒体

(57)【要約】
【課題】　別々に記録された動画像の連続性を保つように再生できるようにする。
【解決手段】　別々に記録されたClip１とClip２を連続再生するとき、Clip１からClip２へと橋渡しの役割をもつBridge Clipが生成される。Bridge Clipは、Clip１からClip２へと切り替わる部分の、Clip１とClip２との、それぞれ対応する部分から構成される。



1

【特許請求の範囲】

【請求項１】　第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成手段と、

前記生成手段により生成された前記第３のＡＶストリームと前記アドレス情報を記録する記録手段とを含むことを特徴とする情報処理装置。

【請求項２】　前記生成手段により生成された前記アドレス情報に含まれる前記第１のＡＶストリームのソースパケットのアライバルタイムスタンプと、前記第３のＡＶストリームの最初に位置するソースパケットのアライバルタイムスタンプは連続しており、かつ、前記生成手段により生成された前記アドレス情報に含まれる前記第２のＡＶストリームのソースパケットのアライバルタイムスタンプと、前記第３のＡＶストリームの最後に位置するソースパケットのアライバルタイムスタンプは連続していることを特徴とする請求項１に記載の情報処理装置。

【請求項３】　前記第３のＡＶストリーム内のソースパケットのアライバルタイムスタンプには、ただ１つの不連続点が存在することを特徴とする請求項２に記載の情報処理装置。

【請求項４】　前記生成手段により生成された前記アドレス情報に含まれる前記第１のＡＶストリームのソースパケットのアドレスの情報で示されるソースパケット以前のＡＶストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、前記アドレスは決定されることを特徴とする請求項２に記載の情報処理装置。

【請求項５】　前記生成手段により生成された前記アドレス情報に含まれる前記第２のＡＶストリームのソースパケットのアドレスの情報で示されるソースパケット以後のＡＶストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、前記アドレスは決定されることを特徴とする請求項２に記載の情報処理装置。

【請求項６】　前記第３のＡＶストリームが記録媒体上で所定の大きさ以上の連続領域に配置されるように、前

2

記第３のＡＶストリームが生成されることを特徴とする請求項２に記載の情報処理装置。

【請求項７】　第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含むことを特徴とする情報処理方法。

【請求項８】　第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項９】　第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップをコンピュータ

に実行させるプログラム。

【請求項１０】　第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームを記録媒体から読み出す第１の読み出し手段と、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を前記記録媒体から読み出す第２の読み出し手段と、

前記第２の読み出し手段により読み出された前記第３のＡＶストリームに関連する情報に基づいて、前記第１の読み出し手段により読み出された前記第１のＡＶストリームから前記第３のＡＶストリームへ再生を切り替え、前記第３のＡＶストリームから前記第２のＡＶストリームへ再生を切り替えて再生する再生手段とを含むことを特徴とする情報処理装置。

【請求項１１】　第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームの記録媒体からの読み出しを制御する第１の読み出し制御ステップと、

前記第３のＡＶストリームの関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報の前記記録媒体からの読み出しを制御する第２の読み出し制御ステップと、

前記第２の読み出し制御ステップの処理で読み出しが制御された前記第３のＡＶストリームに関連する情報に基づいて、前記第１の読み出し制御ステップの処理で読み出しが制御された前記第１のＡＶストリームから前記第３のＡＶストリームへ再生を切り替え、前記第３のＡＶストリームから前記第２のＡＶストリームへ再生を切り替えて再生する再生ステップとを含むことを特徴とする情報処理方法。

【請求項１２】　第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームの記録媒体からの読み出しを制御する第１の読み出し制御ステップと、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報の前記記録媒体からの読み出しを制御する第２の読み出し制御ステップと、

前記第２の読み出し制御ステップの処理で読み出しが制御された前記第３のＡＶストリームに関連する情報に基づいて、前記第１の読み出し制御ステップの処理で読み出しが制御された前記第１のＡＶストリームから前記第３のＡＶストリームへ再生を切り替え、前記第３のＡＶストリームから前記第２のＡＶストリームへ再生を切り替えて再生する再生ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項１３】　第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームの記録媒体からの読み出しを制御する第１の読み出し制御ステップと、

前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報の前記記録媒体からの読み出しを制御する第２の読み出し制御ステップと、

前記第２の読み出し制御ステップの処理で読み出しが制御された前記第３のＡＶストリームに関連する情報に基づいて、前記第１の読み出し制御ステップの処理で読み出しが制御された前記第１のＡＶストリームから前記第３のＡＶストリームへ再生を切り替え、前記第３のＡＶストリームから前記第２のＡＶストリームへ再生を切り替えて再生する再生ステップとをコンピュータに実行させるプログラム。

【請求項１４】　第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、前記第１のＡＶストリームの所定の部分と前記第２のＡＶストリームの所定の部分から構成され、前記第１のＡＶストリームから前記第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームと、前記第３のＡＶストリームに関連する情報として、前記第１のＡＶストリームから前記第３のＡＶストリームに再生が切り替わるタイミングにおける前記第１のＡＶストリームのソースパケットのアドレスの情報と、前記第３のＡＶストリームから前記第２のＡＶストリームに再生が切り替わるタイミングにおける前記第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録されていることを特徴とする記録媒体。

【発明の詳細な説明】

【０００１】
【発明の属する技術分野】本発明は情報処理装置および方法、記録媒体、プログラム、並びに記録媒体に関し、特に、再生区間おける動画像の連続性を保つ情報処理装置および方法、記録媒体、プログラム、並びに記録媒体に関する。
【０００２】
【従来の技術】近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光デイスクに記録するデジタルのAV信号のソース（供給源）としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。
【０００３】ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG（Moving Picture Experts Group）２方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG１ Video、MPEG２ Video、DV方式をはじめとするデジタル記録方式であれば、１度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。
【０００４】しかしながら、このような記録方法は、供給されたビットストリームを１度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。
【０００５】また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、１度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。
【０００６】以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。
【０００７】
【発明が解決しようとする課題】上述したような記録装置において記録媒体に記録されたデータを再生する際、所定のピクチャまで再生し、そのピクチャから時間的に離れた位置に位置するピクチャを続けて再生するといった、いわいるスキップ再生というのがある。スキップ再生を行った際、再生する映像に時間的な連続性が途切れてしまことがあるといった課題があった。
【０００８】本発明はこのような状況に鑑みてなされたものであり、再生区間おける動画像の連続性を保つように再生できるようにすることを目的とする。
【０００９】
【課題を解決するための手段】本発明の第１の情報処理装置は、第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成手段と、生成手段により生成された第３のＡＶストリームとアドレス情報を記録する記録手段とを含むことを特徴とする。
【００１０】前記生成手段により生成されたアドレス情報に含まれる第１のＡＶストリームのソースパケットのアライバルタイムスタンプと、第３のＡＶストリームの最初に位置するソースパケットのアライバルタイムスタンプは連続しており、かつ、生成手段により生成されたアドレス情報に含まれる第２のＡＶストリームのソースパケットのアライバルタイムスタンプと、第３のＡＶストリームの最後に位置するソースパケットのアライバルタイムスタンプは連続しているようにすることができる。
【００１１】前記第３のＡＶストリーム内のソースパケットのアライバルタイムスタンプには、ただ１つの不連続点が存在するようにすることができる。
【００１２】前記生成手段により生成されたアドレス情報に含まれる第１のＡＶストリームのソースパケットのアドレスの情報で示されるソースパケット以前のＡＶストリームのデータ部分が、記録媒体上で所定の大きさ以

上の連続領域に配置されるように、アドレスは決定されるようにすることができる。

【００１３】前記生成手段により生成されたアドレス情報に含まれる第２のＡＶストリームのソースパケットのアドレスの情報で示されるソースパケット以後のＡＶストリームのデータ部分が、記録媒体上で所定の大きさ以上の連続領域に配置されるように、アドレスは決定されるようにすることができる。

【００１４】前記第３のＡＶストリームが記録媒体上で所定の大きさ以上の連続領域に配置されるように、第３のＡＶストリームが生成されるようにすることができる。

【００１５】本発明の第１の情報処理方法は、第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含むことを特徴とする。

【００１６】本発明の第１の記録媒体のプログラムは、第１のＡＶストリームから第のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップを含むことを特徴とする。

【００１７】本発明の第１のプログラムは、第１のＡＶストリームから第のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成する生成ステップをコンピュータに実行させる。

【００１８】本発明の第２の情報処理装置は、第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームを記録媒体から読み出す第１の読み出し手段と、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を記録媒体から読み出す第２の読み出し手段と、第２の読み出し手段により読み出された第３のＡＶストリームに関連する情報に基づいて、第１の読み出し手段により読み出された第１のＡＶストリームから第３のＡＶストリームへ再生を切り替え、第３のＡＶストリームから第２のＡＶストリームへ再生を切り替えて再生する再生手段とを含むことを特徴とする。

【００１９】本発明の第２の情報処理方法は、第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームの記録媒体からの読み出しを制御する第１の読み出し制御ステップと、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報の記録媒体からの読み出しを制御する第２の読み出し制御ステップと、第２の読み出し制御ステップの処理で読み出しが制御された第３のＡＶストリームに関連する情報に基づいて、第１の読み出し制御ステップの処理で読み出しが制御された第１のＡＶストリームから第３のＡＶストリームへ再生を切り替え、第３のＡＶストリームから第２のＡＶストリームへ再生を切り替えて再生する再生ステップとを含むことを特徴とする。

【００２０】本発明の第２の記録媒体のプログラムは、第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームの記録媒体からの読み出しを制御する第１の読み出し制御ステップと、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのア

ドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報の記録媒体からの読み出しを制御する第２の読み出し制御ステップと、第２の読み出し制御ステップの処理で読み出しが制御された第３のＡＶストリームに関連する情報に基づいて、第１の読み出し制御ステップの処理で読み出しが制御された第１のＡＶストリームから第３のＡＶストリームへ再生を切り替え、第３のＡＶストリームから第２のＡＶストリームへ再生を切り替えて再生する再生ステップとを含むことを特徴とする。

【００２１】本発明の第２のプログラムは、第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームの記録媒体からの読み出しを制御する第１の読み出し制御ステップと、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報の記録媒体からの読み出しを制御する第２の読み出し制御ステップと、第２の読み出し制御ステップの処理で読み出しが制御された第３のＡＶストリームに関連する情報に基づいて、第１の読み出し制御ステップの処理で読み出しが制御された第１のＡＶストリームから第３のＡＶストリームへ再生を切り替え、第３のＡＶストリームから第２のＡＶストリームへ再生を切り替えて再生する再生ステップとをコンピュータに実行させる。

【００２２】本発明の第３の記録媒体は、第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームと、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録されていることを特徴とする。

【００２３】本発明の第１の情報処理装置および方法、並びにプログラムにおいては、第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームが生成されるとともに、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報が生成される。

【００２４】本発明の第２の情報処理装置および方法、並びにプログラムにおいては、第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームが記録媒体から読み出され、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報が記録媒体から読み出され、読み出された第３のＡＶストリームに関連する情報に基づいて第１のＡＶストリームから第３のＡＶストリームへ再生が切り替えられ、第３のＡＶストリームから第２のＡＶストリームへ再生が切り替えられて再生される。

【００２５】

【発明の実施の形態】以下に、本発明の実施の形態について、図面を参照して説明する。図１は、本発明を適用した記録再生装置１の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置１は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【００２６】端子１１には、アナログのビデオ信号が、端子１２には、アナログのオーディオ信号が、それぞれ入力される。端子１１に入力されたビデオ信号は、解析部１４とAVエンコーダ１５に、それぞれ出力される。端子１２に入力されたオーディオ信号は、AVエンコーダ１５に出力される。解析部１４は、入力されたビデオ信号からシーンチェンジなどの特徴点を抽出する。

【００２７】AVエンコーダ１５は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ１６に出力する。

【００２８】符号化ビデオストリームは、例えば、MPEG（Moving Picture Expert Group）２方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG１方式により符号化されたオーディオストリームや、ドルビーAC3方式により符号化さ

れたオーディオストリーム等である。マルチプレクサ１６は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ１７を介して多重化ストリーム解析部１８とソースパケッタイザ１９に出力する。

【００２９】多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケッタイザ１９は、入力された多重化ストリームを、そのストリームを記録させる記録媒体１００のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC（誤り訂正）符号化部２０、変調部２１で所定の処理が施され、書き込み部２２に出力される。書き込み部２２は、制御部２３から出力される制御信号に基づいて、記録媒体１００にAVストリームファイルを書き込む（記録する）。

【００３０】デジタルインタフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子１３に入力される。端子１３に入力されたトランスポートストリームの記録方式には、２通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子２４から制御部２３へ入力される。

【００３１】入力トランスポートストリームをトランスペアレントに記録する場合、端子１３に入力されたトランスポートストリームは、多重化ストリーム解析部１８と、ソースパケッタイザ１９に出力される。これ以降の記録媒体１００へAVストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【００３２】入力トランスポートストリームを再エンコードした後に記録する場合、端子１３に入力されたトランスポートストリームは、デマルチプレクサ２６に入力される。デマルチプレクサ２６は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【００３３】デマルチプレクサ２６により抽出されたストリーム（情報）のうち、ビデオストリームはAVデコーダ２７に、オーディオストリームとシステム情報はマルチプレクサ１６に、それぞれ出力される。AVデコーダ２７は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ１５に出力する。AVエンコーダ１５は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ１６に出力する。

【００３４】一方、デマルチプレクサ２６から出力され、マルチプレクサ１６に入力されたオーディオストリームとシステム情報、および、AVエンコーダ１５から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部１８とソースパケットタイザ１９にスイッチ１７を介して出力される。これ以後の記録媒体１００へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【００３５】本実施の形態の記録再生装置１は、AVストリームのファイルを記録媒体１００に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部２３により作成される。制御部２３への入力情報は、解析部１４からの動画像の特徴情報、多重化ストリーム解析部１８からのAVストリームの特徴情報、および端子２４から入力されるユーザからの指示情報である。

【００３６】解析部１４から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関係する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル（ＣＭ）の開始・終了点などの指定情報（マーク）であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

【００３７】多重化ストリーム解析部１８からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関係する情報であり、例えば、AVストリーム内のIピクチャのアドレス情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関係する情報（マーク）などである。

【００３８】端子２４からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

【００３９】制御部２３は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、 AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベース、記録媒体１００の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部２０、変調部２１で処理されて、書き込み部２２へ入力される。書き込み部２２は、制御部２３から出力される制御信号に基づいて、記録媒体１００へデータベースファイルを記録する。

【００４０】上述したアプリケーションデータベース情報についての詳細は後述する。

【００４１】このようにして記録媒体１００に記録されたAVストリームファイル（画像データと音声データのフ

ァイル）と、アプリケーションデータベース情報が再生される場合、まず、制御部２３は、読み出し部２８に対して、記録媒体１００からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部２８は、記録媒体１００からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部２９、ECC復号部３０の処理を経て、制御部２３へ入力される。

【００４２】制御部２３は、アプリケーションデータベース情報に基づいて、記録媒体１００に記録されているPlayListの一覧を端子２４のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部２３へ入力される。制御部２３は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部２８に指示する。読み出し部２８は、その指示に従い、記録媒体１００から対応するAVストリームを読み出し復調部２９に出力する。復調部２９に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部３０の処理を経て、ソースデパケッタイザ３１出力される。

【００４３】ソースデパケッタイザ３１は、記録媒体１００から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ２６に出力できるストリームに変換する。デマルチプレクサ２６は、制御部２３により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ２７に出力する。AVデコーダ２７は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子３２と端子３３から出力する。

【００４４】また、ユーザインタフェースとしての端子２４から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部２３は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体１００からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部２８に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部２３は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部２８に指示する。

【００４５】また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部２３は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部２８に指示する。

【００４６】読み出し部２８は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【００４７】次に、ユーザが、記録媒体１００に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体１００に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Ａという歌番組から歌手Ａの部分を再生し、その後続けて、番組Ｂという歌番組の歌手Ａの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子２４から再生区間の開始点（イン点）と終了点（アウト点）の情報が制御部２３に入力される。制御部２３は、AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベースを作成する。

【００４８】ユーザが、記録媒体１００に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子２４から消去区間のイン点とアウト点の情報が制御部２３に入力される。制御部２３は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書き込み部２２に指示する。

【００４９】ユーザが、記録媒体１００に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部２３は、AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【００５０】まず、端子２４から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部２３へ入力される。制御部２３は、読み出し部２８にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部２８は、記録媒体１００からデータを読み出し、そのデータは、復調部２９、ECC復号部３０、ソースデパケッタイザ３１を経て、デマルチプレクサ２６に出力される。

【００５１】制御部２３は、デマルチプレクサ２６に入力されたデータを解析して、ビデオストリームの再エンコード方法（picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て）と、再多重化方式を決定し、その方式をAVエンコーダ１５とマルチプレクサ１６に供給する。

【００５２】次に、デマルチプレクサ２６は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、「AVデコーダ２７に入力されるデータ」と「マルチプレクサ１６に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータ

であり、これはAVデコーダ２７で復号され、復号されたピクチャはAVエンコーダ１５で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ１６に入力される。

【００５３】マルチプレクサ１６は、制御部２３から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部２０、変調部２１で処理されて、書き込み部２２に入力される。書き込み部２２は、制御部２３から供給される制御信号に基づいて、記録媒体１００にAVストリームを記録する。

【００５４】以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図２は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの２つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlayListの管理をする。ここでは、１つのAVストリームとその付属情報のペアを１つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【００５５】１つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリーポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

【００５６】PlayListについて、図３を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。１つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の１つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点（IN）とアウト点（OUT）の対で表される。従って、PlayListは、複数のPlayItemが集まることにより構成される。

【００５７】PlayListには、２つのタイプがある。１つは、Real PlayListであり、もう１つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、それの参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

【００５８】Virtual PlayListは、Clipのデータを共有していない。従って、Virtual PlayListが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【００５９】次に、Real PlayListの編集について説明する。図４（Ａ）は、Real PlayListのクリエイト(create：作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

【００６０】図４（Ｂ）は、Real PlayListのディバイド(divide：分割)に関する図であり、Real PlayListが所望な点で分けられて、２つのReal PlayListに分割される操作である。この分割という操作は、例えば、１つのPlayListにより管理される１つのクリップ内に、２つの番組が管理されているような場合に、ユーザが１つ１つの番組として登録（記録）し直したいといったようなときに行われる。この操作により、Clipの内容が変更される（Clip自体が分割される）ことはない。

【００６１】図４（Ｃ）は、Real PlayListのコンバイン(combine：結合)に関する図であり、２つのReal PlayListを結合して、１つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが２つの番組を１つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される（Clip自体が１つにされる）ことはない。

【００６２】図５（Ａ）は、Real PlayList全体のデリート(delete：削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

【００６３】図５（Ｂ）は、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【００６４】図５（Ｃ）は、Real PlayListのミニマイズ(Minimize：最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayList にとって不必要なClipの、対応するストリーム部分は削除される。

【００６５】上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

【００６６】そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが

参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか？」といったメッセージなどを表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行、または、キャンセルする。または、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

【００６７】次にVirtual PlayListに対する操作について説明する。Virtual PlayListに対して操作が行われたとしても、Clipの内容が変更されることはない。図６は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

【００６８】図６（Ａ）に示したように、２つのReal PlayList１，２と、それぞれのRealPlayListに対応するClip１，２が存在している場合に、ユーザがReal PlayList１内の所定の区間（In１乃至Out１までの区間：PlayItem１）を再生区間として指示し、続けて再生する区間として、Real PlayList２内の所定の区間（In２乃至Out２までの区間：PlayItem２）を再生区間として指示したとき、図６（Ｂ）に示すように、PlayItem１とPlayItem２から構成される１つのVirtual PlayListが作成される。

【００６９】次に、Virtual PlayList の再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加(append)、Virtual PlayListの中のPlayItemの削除などがある。また、Virtual PlayListそのものを削除することもできる。

【００７０】図７は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

【００７１】Real PlayListとVirtual PlayListで共通の操作として、図８に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList（図２０などを参照して後述する）によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【００７２】次に、マーク（Mark）について説明する。マークは、ClipおよびPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば、シーンチェンジ点などである。PlayListを再生する時、そのPlayListが参照するClipのマークを参照して、使用する事ができる。

【００７３】PlayListに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。ClipまたはPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【００７４】次にサムネイルについて説明する。サムネイルは、Volume、PlayList、およびClipに付加される静止画である。サムネイルには、２つの種類があり、１つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう１つは、マークが指しているシーンを表す画像である。

【００７５】Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体１００、以下、記録媒体１００はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置１の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

【００７６】Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻０の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上２種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、すべてのメニューサムネイルを１つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図１０に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

【００７７】一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必

要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnails）と称する。従って、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

【００７８】図１１は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図１２は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置１がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

【００７９】また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは１つのファイルに格納するのがよい。Playlistはメニューサムネイル１つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

【００８０】図１３は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、およびClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

【００８１】次に、CPI（Characteristic Point Information）について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、２種類のCPIを用いる。１つは、EP_mapであり、もう一つは、TU_mapである。

【００８２】EP_mapは、エントリーポイント(EP)データのリストであり、それはエレメンタリーストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリーポイントの場所を見つけるためのアドレス情報を持つ。１つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【００８３】EP_mapは、主に２つの目的のために使用される。第１に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第２に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置１が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

【００８４】TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置１が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

【００８５】本実施の形態では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号（例えばDV）をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【００８６】SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置１が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

【００８７】デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体１００に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【００８８】あるいは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【００８９】例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

【００９０】デジタル放送のストリームが、記録媒体１００に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力

トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP_mapが作成されてディスクに記録される。

【００９１】または、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にTU_mapが作成されてディスクに記録される。

【００９２】次にディレクトリとファイルについて説明する。以下、記録再生装置１をDVR（Digital Video Recording）と適宜記述する。図１４はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図１４に示したように、"DVR"ディレクトリを含むrootディレクトリ、"PLAYLIST"ディレクトリ、"CLIPINF"ディレクトリ、"M2TS"ディレクトリ、および"DATA"ディレクトリを含む"DVR"ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【００９３】"DVR"ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。"DVR"ディレクトリは、４個のディレクトリを含む。"PLAYLIST"ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが１つもなくても存在する。

【００９４】"CLIPINF"ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが１つもなくても存在する。"M2TS"ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが１つもなくても存在する。"DATA"ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【００９５】"DVR"ディレクトリは、次に示すファイルをストアする。"info.dvr"ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ一つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。"menu.thmb"ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは１つのメニューサムネイルがなければならない。ファイル名は、memu.thmbに固定されるとする。メニューサムネイル画像が１つもない場合、このファイルは、存在しなくても良い。

【００９６】"mark.thmb"ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは１つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が１つもない場合、このファイルは、存在しなくても良い。

【００９７】"PLAYLIST"ディレクトリは、２種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。"xxxxx.rpls"ファイルは、１つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、１つのファイルが作られる。ファイル名は、"xxxxx.rpls"である。ここで、"xxxxx"は、５個の０乃至９まで数字である。ファイル拡張子は、"rpls"でなければならないとする。

【００９８】"yyyyy.vpls"ファイルは、１つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、１つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、５個の０乃至９まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

【００９９】"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、１つのファイルをストアする。"zzzzz.clpi"ファイルは、１つのAVストリームファイル(Clip AV stream file または Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、５個の０乃至９までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

【０１００】"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、５個の０乃至９までの数字である。ファイル拡張子は、"m2ts"でなければならないとする。

【０１０１】STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。仮に、AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻を、PTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。

【０１０２】更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットでは STC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【０１０３】プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ１つのシステムタイムベースを共有するものである。記録再生装置１にとって、AVストリームのデコードに先だち、そのAVストリームの内容がわかることは有

用である。例えば、ビデオやオーディオのエレメンタリーストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど）などの情報である。

【０１０４】この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダおよびデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

【０１０５】MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリーストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化するなどである。

【０１０６】ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceと呼ぶ。Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【０１０７】"DATA"ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

【０１０８】次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。まず、"info.dvr"ファイルについて説明する。図１５は、"info.dvr"ファイルのシンタクスを示す図である。"info.dvr"ファイルは、３個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakerPrivateData()である。

【０１０９】図１５に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１１０】MakerPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word（パディングワード）は、info.dvrのシンタクスに従って挿入される。Ｎ１とＮ２は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【０１１１】DVRVolume()は、ボリューム（ディスク）の内容を記述する情報をストアする。図１６は、DVRVolume()のシンタクスを示す図である。図１６に示したDVRVolume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す４個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

【０１１２】lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

【０１１３】ResumeVolume()は、ボリュームの中で最後に再生したReal PlayListまたはVirtual PlayListのファイル名を記憶している。ただし、Real PlayListまたはは、Virtual PlayListの再生をユーザが中断した時の再生位置は、PlayListMark()において定義されるresume-markにストアされる。

【０１１４】図１７は、ResumeVolume()のシンタクスを示す図である。図１７に示したResumeVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが０にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

【０１１５】resume_PlayList_nameの１０バイトのフィールドは、リジュームされるべきReal PlayListまたはVirtual PlayListのファイル名を示す。

【０１１６】図１６に示したDVRVolume()のシンタクスのなかの、UIAppInfoVolume は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図１８は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの８ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示される値に対応する。

【０１１７】name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【０１１８】Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが１にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事（再生される事）が許可される。このフラグが０にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテン

ツを、ユーザに見せる事が許可される。

【０１１９】最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが０にセットされているか、または、このフラグが１にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置１は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

【０１２０】PINは、４個の０乃至９までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【０１２１】次に図１５に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されているすべてのPlayListファイルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

【０１２２】図２０は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバーを示す４個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０１２３】lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す３２ビットの符号なしの整数である。number_of_PlayListsの１６ビットのフィールドは、PlayList_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。PlayList_file_nameの１０バイトの数字は、PlayListのファイル名を示す。

【０１２４】図２１は、TableOfPlayLists()のシンタクスを別実施の構成を示す図である。図２１に示したシンタクスは、図２０に示したシンタクスに、UIAppinfoPlayList（後述）を含ませた構成とされている。このように、UIAppinfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図２０に示したシンタクスを用いるとして以下の説明をする。

【０１２５】図１５に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置１のメーカが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカのプライベートデータを挿入できるように設けられている。各メーカのプライベートデータは、それを定義したメーカを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、１つ以上のmaker_IDを含んでも良い。

【０１２６】所定のメーカが、プライベートデータを挿入したい時に、すでに他のメーカのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカのプライベートデータが、１つのMakersPrivateData()に含まれることが可能であるようにする。

【０１２７】図２２は、MakersPrivateDataのシンタクスを示す図である。図２２に示したMakersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバを示す４個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す３２ビットの符号なし整数を示す。

【０１２８】mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカプライベートデータのエントリー数を与える１６ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカプライベートデータが２個以上存在してはならない。

【０１２９】mpd_block_sizeは、１０２４バイトを単位として、１つのmpd_blockの大きさを与える１６ビットの符号なし整数である。例えば、mpd_block_size=１ならば、それは１つのmpd_blockの大きさが１０２４バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える１６ビットの符号なし整数である。maker_IDは、そのメーカプライベートデータを作成したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンサによって指定される。

【０１３０】maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバーコ

ードを示す１６ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す１６ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

【０１３１】mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す３２ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中のすべてのmpd_blockは、同じサイズでなければならない。

【０１３２】次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図２３は、xxxxx.rpls（Real PlayList）、または、yyyyy.vpls（Virtual PlayList）のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、３個のオブジェクトから構成され、それらは、PlayList()、PlayListMark()、およびMakerPrivateData()である。

【０１３３】PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１３４】MakerPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１３５】padding_word（パディングワード）は、PlayListファイルのシンタクスにしたがって挿入され、Ｎ１とＮ２は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【０１３６】ここで、既に、簡便に説明したが、PlayListについてさらに説明する。ディスク内にあるすべてのReal PlayListによって、Bridge-Clip（後述）を除くすべてのClipの中の再生区間が参照されていなければならない。かつ、２つ以上のRealPlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

【０１３７】図２４を参照してさらに説明するに、図２４（Ａ）に示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図２４（Ｂ）に示したように、編集作業が行われた後においても守られる。従って、全てのClipは、どれかしらのReal PlayListを参照することにより、必ず視聴することが可能である。

【０１３８】図２４（Ｃ）に示したように、Virtual PlayListの再生区間は、Real PlayListの再生区間またはBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual PlayListにも参照されないBridge-Clipがディスクの中に存在してはならない。

【０１３９】Real PlayListは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual PlayListは、PlayItemのリストを含み、PlayList()の中に示されるCPI_typeがEP_map typeであり、かつPlayList_typeが0（ビデオとオーディオを含むPlayList）である場合、Virtual PlayListは、ひとつのSubPlayItemを含む事ができる。本実施の形態におけるPlayList()では、SubPlayIteはオーディオのアフレコの目的にだけに使用される、そして、１つのVirtual PlayListが持つSubPlayItemの数は、０または１でなければならない。

【０１４０】次に、PlayListについて説明する。図２５は、PlayListのシンタクスを示す図である。図２５に示したPlayListのシンタクスを説明するに、version_numberは、このPlayList()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までのPlayList()のバイト数を示す３２ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す８ビットのフィールドであり、その一例を図２６に示す。

【０１４１】CPI_typeは、１ビットのフラグであり、PlayItem()およびSubPlayItem()によって参照されるClipのCPI_typeの値を示す。１つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す１６ビットのフィールドである。

【０１４２】所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、０から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す１６ビットのフィールドである。この値は、０または１である。付加的なオーディオストリームのパス(オーディオストリームパス)は、サブパスの一種である。

【０１４３】次に、図２５に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインターフェースアプリケーションのパラメータをストアする。図２７は、UIAppInfoPlayListのシンタクスを示す図である。図２７に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、８ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、

図１９に示したテーブルに準拠する値に対応する。

【０１４４】name_lengthは、８ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【０１４５】record_time_and_dateは、PlayListが記録された時の日時をストアする５６ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、１４個の数字を４ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03 は、"0x20011223010203"と符号化される。

【０１４６】durationは、PlayListの総再生時間を時間／分／秒の単位で示した２４ビットのフィールドである。このフィールドは、６個の数字を４ビットのBinary CodedDecimal(BCD)で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【０１４７】valid_periodは、PlayListが有効である期間を示す３２ビットのフィールドである。このフィールドは、８個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、記録再生装置１は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、"0x20010507"と符号化される。

【０１４８】maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置１）の製造者を示す１６ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンサによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す１６ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【０１４９】playback_control_flagのフラグが１にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが０にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

【０１５０】write_protect_flagは、図２８（Ａ）にテーブルを示すように、１にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去および変更されない。このフラグが０にセットされている場合、ユーザは、そのPlayListを自由に消去および変更できる。このフラグが１にセットされている場合、ユーザが、そのPlayListを消去、編集、または上書きする前に、記録再生装置１はユーザに再確認するようなメッセージを表示させる。

【０１５１】write_protect_flagが０にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが１にセットされていても良い。ユーザが、RealPlayListを消去しようとする場合、記録再生装置１は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、または、そのReal PlayListを"Minimize"する。

【０１５２】is_played_flagは、図２８（Ｂ）に示すように、フラグが１にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、０にセットされている場合、そのPlayListは、記録されてから一度も再生されたことがないことを示す。

【０１５３】archiveは、図２８（Ｃ）に示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す２ビットのフィールドである。ref_thumbnail_index のフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

【０１５４】次にPlayItemについて説明する。１つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

【０１５５】PlayListが２つ以上のPlayItemから構成される時、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図２９に示す。

【０１５６】図３０は、PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time（図の中でIN_time1と示されているもの）

は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time（図の中でOUT_time1と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

【０１５７】現在のPlayItemのIN_time（図の中でIN_time2と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time（図の中でOUT_time2と示されているもの）は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

【０１５８】図３１に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

【０１５９】PlayItemのシンタクスは、図３２に示すようになる。図３２に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、ClipInformation fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【０１６０】STC_sequence_idは、８ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この８ビットフィールドは何も意味を持たず、０にセットされる。IN_timeは、３２ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図３３に示すように、PlayList()において定義されるCPI_typeによって異なる。

【０１６１】OUT_timeは、３２ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図３４に示すように、PlayList()において定義されるCPI_typeによって異なる。

【０１６２】Connection_Conditionは、図３５に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す２ビットのフィールドである。図３６は、図３５に示したConnection_Conditionの各状態について説明する図である。

【０１６３】次に、BridgeSequenceInfoについて、図３７を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge_Clip_Information_file_nameを含む。

【０１６４】また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

【０１６５】図３７において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AVstreamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

【０１６６】図３８は、BridgeSequenceinfoのシンタクスを示す図である。図３８に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

【０１６７】RSPN_exit_from_previous_Clipの３２ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【０１６８】RSPN_enter_to_current_Clipの３２ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【０１６９】次に、SubPlayItemについて、図３９を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合だけに許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_inf

ormation_file_ nameを含む。

【０１７０】また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

【０１７１】図４０は、SubPlayItemのシンタクスを示す図である。図４０に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【０１７２】SubPath_typeの８ビットのフィールドは、sub pathのタイプを示す。ここでは、図４１に示すように、'0x00' しか設定されておらず、他の値は、将来のために確保されている。

【０１７３】sync_PlayItem_idの８ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図２５参照）。

【０１７４】sync_start_PTS_of_PlayItemの３２ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentaiotn Time Stamp)の上位３２ビットを示す。SubPath_IN_timeの３２ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する３３ビット長のPTSの上位３２ビットを示す。

【０１７５】SubPath_OUT_timeの３２ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresenation_end_TSの値の上位３２ビットを示す。 Presentation_end_TS = PTS_out + AU_durationここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの９０kHz単位の表示期間である。

【０１７６】次に、図２３に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図４２は、PlayListMarkのシンタクスを示す図である。図４２に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０１７７】lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す３２ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す１６ビットの符号なし整数である。number_of_PlayList_marks は、０であってもよい。mark_typeは、マークのタイプを示す８ビットのフィールドであり、図４３に示すテーブルに従って符号化される。

【０１７８】mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図４４に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する８ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図２５参照）。

【０１７９】character_setの８ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示した値に対応する。name_lengthの８ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【０１８０】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていない事を示す。

【０１８１】次に、Clip information fileについて説明する。zzzzz.clpi（Clip information fileファイル）は、図４５に示すように６個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakerPrivateData()である。AVストリーム(Clip AVストリームまたはBridge-Clip AV stream)とそれに対応するClip Information

ファイルは、同じ数字列の"zzzzz"が使用される。

【０１８２】図４５に示したzzzzz.clpi（Clip information fileファイル）のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１８３】STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１８４】ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakerPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData ()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word（パディングワード）は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。Ｎ１，Ｎ２，Ｎ３，Ｎ４、およびＮ５は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【０１８５】次に、ClipInfoについて説明する。図４６は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル（Clip AVストリームまたは Bridge-Clip AVストリームファイル）の属性情報をストアする。

【０１８６】図４６に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す３２ビットの符号なし整数である。Clip_stream_typeの８ビットのフィールドは、図４７に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【０１８７】offset_SPNの３２ビットのフィールドは、AVストリーム（Clip AVストリームまたは Bridge-Clip AVストリーム）ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは０でなければならない。

【０１８８】図４８に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_xxx（xxxは変形する。例. RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

【０１８９】AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

SPN_xxx = RSPN_xxx - offset_SPN

図４８に、 offset_SPNが、４である場合の例を示す。

【０１９０】TS_recording_rateは、２４ビットの符号なし整数であり、この値は、DVRドライブ（書き込み部２２）へまたはDVRドライブ（読み出し部２８）からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録された時の日時をストアする５６ビットのフィールドであり、年／月／日／時／分／秒について、１４個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【０１９１】durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した２４ビットのフィールドである。このフィールドは、６個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【０１９２】time_controlled_flag:のフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが１である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

TS_average_rate * 192/188 * (t - start_time)－a <= size_clip(t)
<= TS_average_rate * 192/188 * (t - start_time)＋α

ここで、TS_average_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/second の単位で表したものである。

【０１９３】また、上式において、 tは、秒単位で表される時間を示し、start_timeは、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_clip(t)は、 時刻 t におけるAVストリームファイルのサイズをバイト単位で表したも

のであり、例えば、start_timeから時刻tまでに１０個のソースパケットが記録された場合、size_clip(t)は10*192バイトである。aは、TS_average_rateに依存する定数である。

【０１９４】time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

【０１９５】TS_average_rateは、time_controlled_flagが１にセットされている場合、この２４ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが０にセットされている場合、このフィールドは、何も意味を持たず、０にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【０１９６】RSPN_arrival_time_discontinuityの３２ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した SPN_xxx = RSPN_xxx - offset_SPNに基づいて算出される。

【０１９７】reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが１である時、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが１である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが１である場合、transport_stream_IDのフィールドが有効であることを示す。is_servece_ID_validのフラグが１である場合、servece_IDのフィールドが有効であることを示す。

【０１９８】is_ country_code_validのフラグが１である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration deascriotor（ISO/IEC13818-1で定義されている）が持つformat_identifierの値を示す。original_network_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

【０１９９】servece_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているservece_IDの値を示す。country_codeの２４ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x500 x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の１６個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

【０２００】format_identifier、original_network_ID、transport_stream_ID、 servece_ID,country_code 、およびstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

【０２０１】次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図５０は、連続なＳＴＣ区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（ただし、後述するように、Clipの最大時間長は制限されている）。従って、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

【０２０２】STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図５１を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目（k>=0）のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【０２０３】図５２は、STC_Infoのシンタクスを示す図である。図５２に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０２０４】lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す３２ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロをセットしても良い。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

【０２０５】num_of_STC_sequencesの８ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、０から開始される。

【０２０６】RSPN_STC_startの３２ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしても良い。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。

【０２０７】次に、図４５に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図５３を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリーストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【０２０８】program_sequenceは、同一の時刻において、ただ１つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ１つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

【０２０９】図５４は、ProgramInfoのシンタクスを示す図である。図５４に示したProgramInfoのシンタクを説明するに、version_numberは、このProgramInfo()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０２１０】lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す３２ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロにセットされても良い。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは１以上の値でなければならない。

【０２１１】number_of_program_sequencesの８ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは１をセットされなければならない。RSPN_program_sequence_startの３２ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【０２１２】RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

【０２１３】PCR_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの８ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの８ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【０２１４】audio_stream_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフ

ィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【０２１５】なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【０２１６】図５５は、図５４に示したPrograminfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図５５に示したVideoCodingInfoのシンタクスを説明するに、video_formatの８ビットフィールドは、図５６に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

【０２１７】frame_rateの８ビットフィールドは、図５７に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの８ビットフィールドは、図５８に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

【０２１８】図５９は、図５４に示したPrograminfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図５９に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、図６０に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

【０２１９】audio_component_typeの８ビットフィールドは、図６１に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの８ビットフィールドは、図６２に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

【０２２０】次に、図４５に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには２つのタイプがあり、それらはEP_mapとTU_mapである。図６３に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図６４に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。１つのAVストリームは、１つのEP_mapまたは一つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

【０２２１】図６５は、CPIのシンタクスを示す図である。図６５に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す４個のキャラクター文字である。version_numberは、ISO646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す３２ビットの符号なし整数である。CPI_typeは、図６６に示すように、１ビットのフラグであり、ClipのCPIのタイプを表す。

【０２２２】次に、図６５に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、２つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが１つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、１つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

【０２２３】ビデオストリーム用のEP_mapについて図６７を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、および、RSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始めるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第１バイト目を含むソースポケットのアドレスを示す。

【０２２４】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【０２２５】オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第１バイト目を含むソースポケットのアドレスを示す。

【０２２６】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【０２２７】EP_mapとSTC_Infoの関係を説明するに、１つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく１つのテーブルに作られる。RSPN_EP_startの

値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図６８を参照）。・EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、１つのEP_map_for_one_stream_PIDを持たねばならない。図６９に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

【０２２８】図７０は、EP_mapのシンタクスを示す図である。図７０に示したEP_mapのシンタクスを説明するに、EP_typeは、４ビットのフィールドであり、図７１に示すように、EP_mapのエントリーポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが１つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、１つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

【０２２９】number_of_stream_PIDsの１６ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の１６ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム（ビデオまたはオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが0 ('video')に等しい場合、そのエレメンタリストリームはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

【０２３０】num_EP_entries(k)の１６ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k)：この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第１バイト目からの大きさで示される。

【０２３１】padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【０２３２】図７２は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図７２に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの３２ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの３３ビット精度のPTSの上位３２ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの３３ビット精度のPTSの上位３２ビットを持つ。

【０２３３】RSPN_EP_startの３２ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第１バイト目を含むソースパケットの相対アドレスを示す。または、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第一バイト目を含むソースパケットの相対アドレスを示す。

【０２３４】RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

【０２３５】次に、TU_mapについて、図７３を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、１つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

【０２３６】AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

TU_start_time(k) = offset_time + k*time_unit_size

TU_start_time(k)は、45kHzの精度を持つ。

【０２３７】図７５は、TU_mapのシンタクスを示す図である。図７５に示したTU_mapのシンタクスを説明するに、offset_timeの３２bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、２７MHz精度のアライバルタイムクロックから導き出される４５kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeはゼロにセットされなけ

ればならない。

【０２３８】time_unit_sizeの３２ビットフィールドは、time_unitの大きさを与えるものであり、それは２７MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、１秒以下（time_unit_size<=45000）にすることが良い。number_of_time_unit_entriesの３２ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリー数を示す。

【０２３９】RSPN_time_unit_startの３２ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

【０２４０】図４５に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置１）によってセットされるものであり、ユーザによってセットされるものではない。

【０２４１】図７５は、ClipMarkのシンタクスを示す図である。図７５に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０２４２】lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す３２ビットの符号なし整数である。number_of_Clip_marksは、 ClipMarkの中にストアされているマークの個数を示す１６ビットの符号なし整数。number_of_Clip_marks は、０であってもよい。mark_typeは、マークのタイプを示す８ビットのフィールドであり、図７６に示すテーブルに従って符号化される。

【０２４３】mark_time_stampは、３２ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図７７に示すように、PlayList()の中のCPI_typeにより異なる。

【０２４４】STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この８ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この８ビットのフィールドは何も意味を持たず、ゼロにセットされる。character_setの８ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示される値に対応する。

【０２４５】name_lengthの８ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【０２４６】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていない。

【０２４７】MakersPrivateDataについては、図２２を参照して既に説明したので、その説明は省略する。

【０２４８】次に、サムネイルインフォメーション（Thumbnail Information）について説明する。サムネイル画像は、menu.thmbファイルまたはmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ１つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像，すなわちVolumeを代表する画像、および、それぞれのPlayListを代表する画像をストアする。すべてのメニューサムネイルは、ただ１つのmenu.thmbファイルにストアされる。

【０２４９】mark.thmbファイルは、マークサムネイル画像，すなわちマーク点を表すピクチャをストアする。すべてのPlayListおよびClipに対するすべてのマークサムネイルは、ただ１つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は一つのtn_blockに格納される。１つの画像データはは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。１つのサムネイル画像のバイト長は可変である。

【０２５０】図７８は、menu.thmbとmark.thmbのシンタクスを示す図であり、図７９は、図７８に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタク

スを示す図である。図７９に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０２５１】lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す３２ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す３２ビットの符号なし整数である。相対バイト数はゼロからカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリー数を与える１６ビットの符号なし整数である。

【０２５２】tn_block_sizeは、1024バイトを単位として、１つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size＝１ならば、それは１つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す１１６ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデクス番号を表す１６ビットの符号なし整数である。thumbnail_index として、0xFFFFという値を使用してはならない。thumbnail_index はUIAppInfoVolume()、UIAppInfoPlayList()、 PlayListMark()、およびClipMark()の中のref_thumbnail_indexによって参照される。

【０２５３】thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図８０に示すような値をとる。表中のDCFとPNGは"menu.thmb"内でのみ許される。マークサムネイルは、値"0x00" (MPEG-2 Video I-picture)をとらなければならない。

【０２５４】picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す３２ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す１６ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。tn_block番号は、０から始まり、tn_blockのfor-ループ中の変数ｋの値に関係する。

【０２５５】x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す１６ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す１６ビットの符号なし整数である。tn_blockは、 サムネイル画像がストアされる領域である。Thumbnail()の中のすべてのtn_blockは、同じサイズ（固定長）であり、その大きさはtn_block_sizeによって定義される。

【０２５６】図８１は、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図８１のように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応する事ができるようになる。

【０２５７】次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ（図１４）にストアされる。AVストリームファイルには、２つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【０２５８】まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図８２に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第１バイト目から始まる。ソースパケットは、192バイト長である。一つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、４バイト長であり、またトランスポートパケットは、188バイト長である。

【０２５９】１つのAligned unitは、３２個のソースパケットから成る。DVR MPEG２トランスポートストリームの中の最後のAligned unitも、また３２個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が３２の倍数でない時、ヌルパケット（PID=0x1FFFのトランスポートパケット）を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG２トランスポートストリームに余分な情報を付加してはならない。

【０２６０】図８３に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図８３に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【０２６１】MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG2トランスポートストリームは、フルトランスポートストリームまたはパ

ーシャルトランスポートストリームである。入力されるMPEG2トランスポートストリームは、ISO/IEC13818-1またはISO/IEC13818-9に従っていなければならない。MPEG2トランスポートストリームのi番目のバイトは、T-STD(ISO/IEC 13818-1で規定されるTransport stream system target decoder)とソースパケッタイザーへ、時刻t(i)に同時に入力される。Rpkは、トランスポートパケットの入力レートの瞬時的な最大値である。

【０２６２】２７MHz PLL５２は、２７MHzクロックの周波数を発生する。２７MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。arrival time clock counter５３は、２７MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

【０２６３】source packetizer５４は、すべてのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第１バイト目がT-STDとソースパケッタイザーの両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第１バイト目を示す。

arrival_time_stamp(k) = arrival_time_clock(k)% 2^{30}

【０２６４】２つの連続して入力されるトランスポートパケットの時間間隔が、2^{30} /27000000秒（約40秒）以上になる場合、その２つのトランスポートパケットのarrival_time_stampの差分は、2^{30} /27000000秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

【０２６５】smoothing buffer５５は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファは、オーバーフロウしてはならない。Rmaxは、スムージングバッファが空でない時のスムージングバッファからのソースパケットの出力ビットレートである。スムージングバッファが空である時、スムージングバッファからの出力ビットレートはゼロである。

【０２６６】次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

Rmax = TS_recording_rate * 192/188

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

【０２６７】入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター，例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照しても良い。

【０２６８】smoothing buffer sizeは、入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファの大きさはゼロである。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファの大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照しても良い。

【０２６９】記録機（レコーダ）および再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォールトのバッファサイズは、1536 bytesである。

【０２７０】次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図８４は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【０２７１】27MHz X-tal６１は、２７Mhzの周波数を発生する。２７MHz周波数の誤差範囲は、+/-30 ppm (27000000 +/- 810 Hz)でなければならない。arrival time clock counter６２は、２７MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

【０２７２】smoothing buffer６４において、Rmaxは、スムージングバッファがフルでない時のスムージングバッファへのソースパケットの入力ビットレートである。スムージングバッファがフルである時、スムージングバッファへの入力ビットレートはゼロである。

【０２７３】MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB ３０ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スムージングバッファから引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファは、アンダーフロウしてはならない。

【０２７４】DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

【０２７５】図８５は、Source packetのシンタクスを

示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図８５に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図８６に示す。図８６に示したTP_Extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、またはcopy prohibitedとすることができる。図８７は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

【０２７６】copy_permission_indicatorは、すべてのトランスポートパケットに付加される。IEEE1394デジタルインターフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronouspacket headerの中のEMI(Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【０２７７】arrival_time_stampは、次式
arrival_time_stamp(k) = arrival_time_clock(k)% 2^{30}
において、arrival_time_stampによって指定される値を持つ整数値である。

【０２７８】Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース（STCベース）の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

【０２７９】Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、２６時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース（STCベース）の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間).と規定している。

【０２８０】Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、１つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならず、かつ後述するDVR-STDに従わなければならない。

【０２８１】本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ／レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフロウを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

【０２８２】"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせる事なく、ディスクに記録されたオーディオビデオデータを表示できることである。

【０２８３】シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

【０２８４】図８８は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図８８においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図８８に示したTS1は、Clip1（Clip AVストリーム）の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータから成る。

【０２８５】TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図８８においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【０２８６】また、図８８におけるTS2は、Clip2（Clip AVストリーム）の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含ま

れるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time（図８８においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【０２８７】図８９は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図８９におけるTS1は、Clip1（Clip AVストリーム)の影を付けられたストリームデータから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図８９においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図８９におけるTS2は、Clip2(Clip AVストリーム)の影を付けられたストリームデータから成る。

【０２８８】TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time（図８９においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【０２８９】図８８と図８９において、TS1とT2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、１でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、１でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、２以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および／またはTS2の中に、上記以外のエレメンタリーストリームまたはプライベートストリームが含まれていても良い。

【０２９０】ビデオビットストリームの制限について説明する。図９０は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1（Clip1のOUT_time）の後とIN_time2（Clip2のIN_time）の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【０２９１】図９０に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図９１に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図９０のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、１つの連続でMPEG２規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【０２９２】同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図９０のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始する事ができて、これに続くClip2のビデオストリームに接続され、１つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【０２９３】図９０に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図９２に示す。Clip1のビデオストリームは、図９０のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、１つの連続でMPEG２規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図９０のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、一つの連続でMPEG２規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【０２９４】ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【０２９５】ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレームまたはフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレームまたはフィールドのギャップがあってはならない。接続点において、トップ?ボトムのフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、"top_field_first" および "repeat_first_field"フラグを書き換える必要があるかもしれない，またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにし

ても良い。

【０２９６】オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例．MPEG1レイヤ2，AC-3，SESF LPCM，AAC）は、同じでなければならない。

【０２９７】次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【０２９８】接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図９３に示すように、２オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあっても良い。TS2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

【０２９９】ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

【０３００】以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-ClipAVストリームは、ただ１つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【０３０１】BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【０３０２】PlayItemの制限について説明するに、先行するPlayItemのOUT_time（図８８、図８９において示されるOUT_time1）は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time（図８８、図８９において示されるIN_time2）は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【０３０３】Bridge-Clipを使用する場合のデータアロケーションの制限について、図９４を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）とClip2（Clip AVストリームファイル）に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【０３０４】RSPN_exit_from_previous_Clip以前のClip1（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

【０３０５】Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図９５を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）の最後の部分とClip2（Clip AVストリームファイル）の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

【０３０６】Clip1（Clip AVストリームファイル）の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2（Clip AVストリームファイル）の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

【０３０７】所定のビットレートを持つディジタルＡＶ信号が、ディスク上に断片化して記録されている場合、記録されたディジタルＡＶ信号を所定のビットレートで記録媒体１００から読み出せることを保証するためには、１つの連続記録領域の大きさが次の条件を満たさなければならない。

S*8 / (S*8/Rud + Ts) >= Rmax

ここで、

S：１つの連続記録領域の最小の大きさ［Byte］

Ts：１つの記録領域から次の記録領域へのフルストロークのアクセス時間[second]

Rud：記録メディアからの読み出しビットレート［bit/second］

Rmax：AVストリームのビットレート［bit/second］

すなわち、ディスク上で、Sバイト以上にAVストリーム

のデータが連続して記録されるようにデータを配置しなければならない。

【０３０８】上記のハーフフラグメントの大きさが、Sバイト以上となるようにデータを配置しなければならない。

【０３０９】次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランスポートストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルでもある。

【０３１０】DVR-STDモデルを図９６に示す。図９６に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n, TBn,MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, OnおよびPn(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデクス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファでる。

【０３１１】MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【０３１２】Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【０３１３】DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB1, TBnまたはTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsysおよびBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【０３１４】シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される２つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図８８に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【０３１５】図９７は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランスポートパケットの入力，復号，表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図９７においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図９７においてATC1で示される）と同じでない。

【０３１６】また、TS2のシステムタイムベースの時間軸（図９７においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図９７においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあっても良い。

【０３１７】DVR-STD への入力タイミングについて説明する。時刻T_1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn またはTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

【０３１８】TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBnまたはTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T_2である。従って、時刻T_1からT_2までの区間では、ソースパケットのarrival_time_stampは無視される。

【０３１９】N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケットのバイト数とすると、時刻T_1乃至T_2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$\Delta T1 = T_2 - T_1 = N1 / TS\_recording\_rate\ (TS1)$

時刻T_1乃至T_2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

【０３２０】T_2の時刻において、arrival time clock counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1, TBn またはTBsysのバッファへの入力タイミングは、TS2のソース

パケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

【０３２１】付加的なオーディオバッファリングおよびシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻Ｔ１からT2までの区間の入力データを処理することができるように、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

【０３２２】ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図９７ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図９７ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

【０３２３】STC1とSTC2の間のオフセットは、次のように決定される。PTS^{1}_{end} は、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS^{2}_{start} は、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、T_{pp} は、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、２つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$STC\_delta = PTS^{1}_{end} + T_{pp} - PTS^{2}_{start}$$

【０３２４】オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは０乃至２オーディオフレーム未満である（図９７に図示されている"audio overlap"を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

【０３２５】DVR-STDのシステムタイムクロックについて説明するに、時刻T_5において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T_2からT_5の間にオーバーラップしていても良い。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値（STC1）と新しいタイムベースの値（STC2）の間で切り替える。STC2の値は、次式により算出される。

STC2＝STC1－STC_delta

【０３２６】バッファリングの連続性について説明する。$STC1^{1}_{video\_end}$ は、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC1上のSTCの値である。$STC2^{2}_{video\_start}$ は、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC2上のSTCの値である。$STC2^{1}_{video\_end}$ は、$STC1^{1}_{video\_end}$ の値をシステムタイムベースSTC2上の値に換算した値である。$STC2^{1}_{video\_end}$ は、次式により算出される。

$$STC2^{1}_{video\_end} = STC1^{1}_{video\_end} - STC\_delta$$

【０３２７】DVR-STDに従うために、次の２つの条件を満たす事が要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC2^{2}_{video\_start} > STC2^{1}_{video\_end} + \Delta T_1$$

この不等式が満たされるように、Clip１および、または、Clip２の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【０３２８】次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフロウおよびアンダーフローさせてはならない。

【０３２９】図９８は、BridgeSequenceInfo()のシンタクスの別例を示す図である。図３８のBridgeSequenceInfo()との違いは、Bridge_Clip_Information_file_nameしか含まれないことである。

【０３３０】図９９は、図９８のBridgeSequenceInfo()のシンタクスを使用する場合、２つのPlayItemが、シームレスに接続される時のBridge-Clipについて説明する図である。RSPN_exit_from_previous_Clipは、先行するPlayItemが参照するClip AVstream上のソースパケットのソースパケット番号であり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。

【０３３１】RSPN_enter_to_current_Clipは、現在のPlayItemが参照するClip AV stream上のソースパケットの番号であり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。図９９に示すBridge-Clip AVストリームファイルにおいて、SPN_ATC_startは、Bridge-Clip AVストリームファイルの中で新しいアライバルタイムベースの時間軸が開始するソースパケットのソースパケット番号を示す。

【０３３２】Bridge-Clip AVストリームファイルは１個のアライバルタイムベースの不連続点を持つ。図中で２番目のSPN_ATC_startは、図３７のRSPN_arrival_time_discontinuity と同じ意味を持つ。

【０３３３】図９８のBridgeSequenceInfo()のシンタクスを使用する場合、RSPN_exit_from_previous_ClipとRSPN_enter_to_current_Clipは、Bridge-Clip AVストリームファイルに対応するClip Informationファイルの中に

ストアされる。また、SPN_ATC_startもまたClip Information ファイルの中にストアされる。

【０３３４】図１００は、BridgeSequenceInfoが、図９８のシンタクスの場合のClip Informationファイルのシンタクスを示す図である。SequenceInfo_start_addressは、Clip Informationファイルの先頭のバイトからの相対バイト数を単位として、SequenceInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０３３５】図１０１は、図１００のClip Informationファイルの ClipInfo()のシンタクスを示す図である。Clip_stream_typeは、そのClipのAVストリームファイルがClipAVストリームファイルであるか、それともBridge-Clip AVストリームファイルであるかを示す。Clip_stream_typeがBridge-Clip AVストリームファイルを示す場合、次のシンタクスフィールドが続く。

【０３３６】previous_Clip_Information_file_nameは、そのBridge-Clip AVストリームファイルの前に接続されるClipのClip Informationファイル名を示す。RSPN_exit_from_previous_Clipは、previous_Clip_Information_file_nameで示されるClipAVストリームファイル上のソースパケットのソースパケット番号であり、そのソースパケットに続いてBridge-Clip AVストリームファイルの最初のソースパケットが接続される。そのソースパケット番号は、Clip AVストリームファイルの最初のソースパケットからゼロを初期値としてカウントされる値である。

【０３３７】current_Clip_Information_file_nameは、そのBridge-Clip AVストリームファイルの後ろに接続されるClipのClip Informationファイル名を示す。RSPN_enter_to_current_Clipは、current_Clip_Information_file_nameで示されるClip AVストリームファイル上のソースパケットのソースパケット番号であり、そのソースパケットの前にBridge-Clip AVストリームファイルの最後のソースパケットが接続される。そのソースパケット番号は、Clip AVストリームファイルの最初のソースパケットからゼロを初期値としてカウントされる値である。

【０３３８】図１０２は、図１００のClip Information ファイルのSequenceInfo()のシンタクスを示す。num_of_ATC_sequencesは、AVストリームファイルの中にあるATC-sequenceの数を示す。ATC-sequenceは、アライバルタイムベースの不連続点を含まないソースパケット列である。Bridge-Clipの場合、この値は２である。

【０３３９】SPN_ATC_start[atc_id] は、AVストリームファイル上でatc_idによって指されるアライバルタイムベースが開始するアドレスを示す。SPN_ATC_start[atc_id]は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからゼロを初期値としてカウントされる。

【０３４０】図１０３は、Bridge-Sequenceによって参照されるClip AVストリームファイルのストリームデータを部分的に消去した場合のデータベースの変更について説明する図である。図１０３（Ａ）の"Before Editing"で示ように、Clip1とClip2がBridge-Clipで接続されていて、RSPN_exit_from_previous_Clip=X, RSPN_exit_from_previous_Clip=Yであるとする。

【０３４１】この時、Clip1の斜線で示すZ1個のソースパケットのストリームデータ部分、およびClip2の斜線で示すZ2個のソースパケットのストリームデータ部分を消去するとする。その結果、図１０３（Ｂ）の"After Editing"で示すように、RSPN_exit_from_previous_Clip=X-Z1, RSPN_exit_from_previous_Clip=Y-Z2 に値が変更される。

【０３４２】BridgeSequenceに関係のあるデータベースのシンタクスを図９８と図１０１のように変更することにより、AVストリーム中のデータアドレスを示すところのソースパケット番号についての情報（データベースのシンタクス中で、RSPNで始まるフィールド）がPlayListのレイヤからなくなり、ソースパケット番号の情報はすべてClipのレイヤで記述されることになる。

【０３４３】これにより、AVストリーム中のデータアドレスの値に変更が必要になった場合（例えばAVストリームファイルのデータを部分的に消去した時にこれが必要になる）、Clipインフォメーションファイルだけをデータ管理すれば良いので、データベースの管理が容易になるメリットがある。

【０３４４】図１０４は、Real PlayListの作成について説明するフローチャートである。図１の記録再生装置１のブロック図を参照しながら説明する。ステップＳ１０において、制御部２３はClip AVストリームを記録する。ステップＳ１１において、制御部２３は、上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlayList()を作成する。Clipの中にSTC不連続点があり、PlayList()が２つ以上のPlayItemからなる場合、PlayItem間のconnection_conditionもまた決定される。

【０３４５】ステップＳ１２において、制御部２３は、UIAppInfoPlayList()を作成する。ステップＳ１３において、制御部２３は、PlayListMarkを作成する。ステップ１４において、制御部２３は、MakersPrivateDataを作成する。ステップＳ１５において、制御部２３は、Real PlayListファイルを記録する。このようにして、新規にClip AVストリームを記録する毎に、１つのReal PlayListファイルが作られる。

【０３４６】図１０５は、ブリッジシーケンスを持つVirtual PlayListの作成について説明するフローチャートである。ステップＳ２０において、ユーザーインターフェースを通して、ディスクに記録されている１つのReal PlayListの再生が指定される。そして、そのReal PlayListの再生範囲の中から、ユーザーインターフェースを通して、IN点とOUT点で示される再生区間が指定され

る。

【０３４７】ステップＳ２１において、制御部２３は、ユーザによる再生範囲の指定操作がすべて終了したか否かを判断し、終了していると判断した場合、ステップＳＳ２２に進み、終了していないと判断した場合、ステップＳ２０に戻り、それ以降の処理が繰り返される。

【０３４８】ステップＳ２２において、連続して再生される２つのPlayItem間の接続状態(connection_condition)を、ユーザがユーザーインタフェースを通して決定するか、または制御部２３が決定する。ステップＳ２３において、制御部２３は、シームレス接続されるPlayItemのためのブリッジシーケンスを作成する。ステップＳ２４において、制御部２３は、Virtual PlayListファイルを作成し、記録する。

【０３４９】図１０６は、ステップＳ２３における詳細な処理を説明するフローチャートである。ステップＳ３１において、制御部２３は、時間的に前側に表示されるPlayItemのOUT点側のAVストリームの再エンコードおよび再多重化を行う。ステップＳ３２において、制御部２３は、上記PlayItemに続いて表示されるPlayItemのIN点側のAVストリームの再エンコードおよび再多重化を行う。

【０３５０】ステップＳ３３において、制御部２３は、データの連続供給のためのデータアロケーション条件を満たすように、RSPN_exit_from_previous_Clipの値を決定する。すなわち、RSPN_exit_from_previous_Clip以前のClip AVストリームファイルのストリーム部分が、記録媒体上で前述のハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない（図９１、図９４を参照）。

【０３５１】ステップＳ３４において、制御部２３は、データの連続供給のためのデータアロケーション条件を満たすように、RSPN_enter_to_current_Clipの値を決定する。すなわち、RSPN_enter_to_current_Clip以後のClip AVストリームファイルのストリーム部分が、記録媒体１００上で前述のハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない（図９１、図９４を参照）。

【０３５２】ステップＳ３５において、制御部２３は、データの連続供給のためのデータアロケーション条件を満たすように、Bridge-Clip AVストリームファイルを作成する。すなわち、ステップＳ３１とステップＳ３２の処理で作成されたデータの量が、前述のハーフフラグメント以上のサイズ未満である場合、オリジナルのClipからデータがコピーされてBridge-Clipが作成される（図９１、図９４を参照）。

【０３５３】ステップＳ３３，Ｓ３４，Ｓ３５の各処理は、時系列に説明しているが、これらの処理はお互いが関係するので、順不同もしくは同時に処理が行われても良い。

【０３５４】ステップＳ３６において、制御部２３は、ブリッジシーケンスのデータベースを作成する。ステップＳ３７において、制御部２３は、Bridge-Clip AVストリームファイルおよびそのClipインフォメーションファイルを記録する。このようにして、ディスクに記録されているReal PlayListの再生範囲の中から、ユーザにより１つ以上のPlayItemが選択され、２つのPlayItem間がシームレス接続できるためのブリッジシーケンスが作成され、１つ以上のPlayItemがグループ化されたものを、１つのVirtual PlayListファイルとして記録される。

【０３５５】図１０７は、PlayListの再生について説明するフローチャートである。ステップＳ４１において、制御部２３は、Info.dvr, Clip Information file, PlayList fileおよびサムネールファイルの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザーインタフェースを通して、GUIに表示する。

【０３５６】ステップＳ４２において、制御部２３は、それぞれのPlayListのUIAppInfoPlayList()に基づいて、PlayListを説明する情報をGUI画面に提示する。ステップＳ４３において、ユーザーインタフェースを通して、GUI画面上からユーザーが１つのPlayListの再生を指示する。ステップＳ４４において、制御部23は、現在のPlayItemのSTC-sequence-idとIN_timeのPTSから、IN_timeより時間的に前で最も近いエントリーポイントのあるソースパケット番号を取得する。

【０３５７】ステップＳ４５において、制御部２３は、上記エントリーポイントのあるソースパケット番号からAVストリームのデータを読み出し、デコーダへ供給する。ステップＳ４６において、現在のPlayItemの時間的に前のPlayItemがあった場合、制御部２３は、前のPlayItemと現在のPlayItemとの表示の接続処理をconnection_conditionに従って行う。PlayItemがシームレス接続される場合、DVR-STDのデコード方法に基づいてＡＶストリームをデコードする。

【０３５８】ステップＳ４７において、制御部２３は、AVデコーダ２７にIN_timeのPTSのピクチャから表示を開始するように指示する。ステップＳ４８において、制御部２３は、AVデコーダ２７にAVストリームのデコードを続けるように指示する。ステップＳ４９において、制御部２３は、現在表示の画像が、OUT_timeのPTSの画像か否かを判断し、OUT_timeのPTSの画像ではないと判断された場合、ステップＳ５０に進み、画像が表示された後、ステップＳ４８に戻り、それ以降の処理が繰り返される。

【０３５９】一方、ステップＳ４９において、現在表示の画像が、OUT_timeのPTSの画像であると判断された場合、ステップＳ５１へ進む。ステップＳ５１において、

制御部２３は、現在のPlayItemがPlayListの中で最後のPlayItemか否かを判断し、最後のPlayItemではないと判断された場合、ステップＳ４４に戻り、それ以降の処理が繰り返され、最後のPlayItemであると判断された場合、PlayListの再生を終了する。

【０３６０】このようにして、ユーザにより再生指示された１つのPlayListファイルの再生が行なわれる。

【０３６１】このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【０３６２】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【０３６３】図１０８は、汎用のパーソナルコンピュータの内部構成例を示す図である。パーソナルコンピュータのCPU（Central Processing Unit）２０１は、ROM（ReadOnly Memory）２０２に記憶されているプログラムに従って各種の処理を実行する。RAM（Random Access Memory）２０３には、CPU２０１が各種の処理を実行する上において必要なデータやプログラムなどが適宜記憶される。入出力インタフェース２０５は、キーボードやマウスから構成される入力部２０６が接続され、入力部２０６に入力された信号をCPU２０１に出力する。また、入出力インタフェース２０５には、ディスプレイやスピーカなどから構成される出力部２０７も接続されている。

【０３６４】さらに、入出力インタフェース２０５には、ハードディスクなどから構成される記憶部２０８、および、インターネットなどのネットワークを介して他の装置とデータの授受を行う通信部２０９も接続されている。ドライブ２１０は、磁気ディスク２２１、光ディスク２２２、光磁気ディスク２２３、半導体メモリ２２４などの記録媒体からデータを読み出したり、データを書き込んだりするときに用いられる。

【０３６５】この記録媒体は、図１０８に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク２２１（フロッピディスクを含む）、光ディスク２２２（CD-ROM（Compact Disk-Read Only Memory），DVD（Digital Versatile Disk）を含む）、光磁気ディスク２２３（MD（Mini-Disk）を含む）、若しくは半導体メモリ２２４などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM２０２や記憶部２０８が含まれるハードディスクなどで構成される。

【０３６６】なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【０３６７】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【０３６８】

【発明の効果】以上の如く、本発明の情報処理装置および方法、並びにプログラムによれば、第１のＡＶストリームから第２のＡＶストリームへ連続的に再生されるように指示された場合、第１のＡＶストリームの所定の部分と第２のＡＶストリームの所定の部分から構成され、第１のＡＶストリームから第２のＡＶストリームに再生が切り換えられるとき再生される第３のＡＶストリームを生成するとともに、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を生成するようにしたので、別々に記録されたＡＶストリームの連続性を保つように再生できる。

【０３６９】また、本発明の第２の情報処理装置および方法、並びにプログラムによれば、第１のＡＶストリーム、第２のＡＶストリーム、または、第３のＡＶストリームを記録媒体から読み出し、第３のＡＶストリームに関連する情報として、第１のＡＶストリームから第３のＡＶストリームに再生が切り替わるタイミングにおける第１のＡＶストリームのソースパケットのアドレスの情報と、第３のＡＶストリームから第２のＡＶストリームに再生が切り替わるタイミングにおける第２のＡＶストリームのソースパケットのアドレスの情報から構成されるアドレス情報を記録媒体から読み出し、読み出された第３のＡＶストリームに関連する情報に基づいて第１のＡＶストリームから第３のＡＶストリームへ再生を切り替え、第３のＡＶストリームから第２のＡＶストリームへ再生を切り替えて再生するようにしたので、別々に記録されたＡＶストリームの連続性を保つように再生できる。

【図面の簡単な説明】

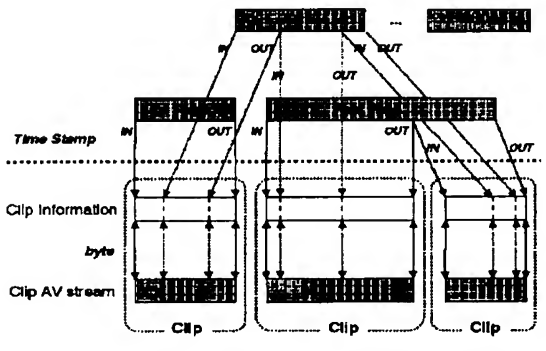
【図１】本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図２】記録再生装置１により記録媒体に記録されるデ

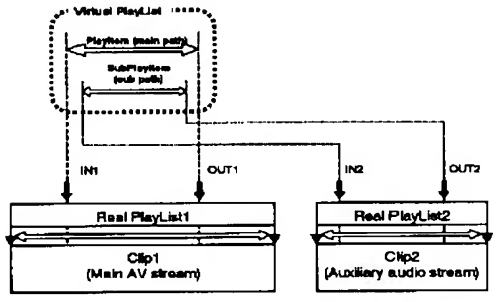
ータのフォーマットについて説明する図である。
【図３】Real PlayListとVirtual PlayListについて説明する図である。
【図４】Real PlayListの作成について説明する図である。
【図５】Real PlayListの削除について説明する図である。
【図６】アセンブル編集について説明する図である。
【図７】Virtual PlayListにサブパスを設ける場合について説明する図である。
【図８】PlayListの再生順序の変更について説明する図である。
【図９】PlayList上のマークとClip上のマークについて説明する図である。
【図１０】メニューサムネイルについて説明する図である。
【図１１】PlayListに付加されるマークについて説明する図である。
【図１２】クリップに付加されるマークについて説明する図である。
【図１３】PlayList、Clip、サムネイルファイルの関係について説明する図である。
【図１４】ディレクトリ構造について説明する図である。
【図１５】info.dvrのシンタクスを示す図である。
【図１６】DVR volumeのシンタクスを示す図である。
【図１７】Resumevolumeのシンタクスを示す図である。
【図１８】UIAppInfovolumeのシンタクスを示す図である。
【図１９】Character set valueのテーブルを示す図である。
【図２０】TableOfPlayListのシンタクスを示す図である。
【図２１】TableOfPlayListの他のシンタクスを示す図である。
【図２２】MakersPrivateDataのシンタクスを示す図である。
【図２３】xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。
【図２４】PlayListについて説明する図である。
【図２５】PlayListのシンタクスを示す図である。
【図２６】PlayList_typeのテーブルを示す図である。
【図２７】UIAppinfoPlayListのシンタクスを示す図である。
【図２８】図２７に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。
【図２９】PlayItemについて説明する図である。
【図３０】PlayItemについて説明する図である。
【図３１】PlayItemについて説明する図である。
【図３２】PlayItemのシンタクスを示す図である。
【図３３】IN_timeについて説明する図である。
【図３４】OUT_timeについて説明する図である。
【図３５】Connection_Conditionのテーブルを示す図である。
【図３６】Connection_Conditionについて説明する図である。
【図３７】BridgeSequenceInfoを説明する図である。
【図３８】BridgeSequenceInfoのシンタクスを示す図である。
【図３９】SubPlayItemについて説明する図である。
【図４０】SubPlayItemのシンタクスを示す図である。
【図４１】SubPath_typeのテーブルを示す図である。
【図４２】PlayListMarkのシンタクスを示す図である。
【図４３】Mark_typeのテーブルを示す図である。
【図４４】Mark_time_stampを説明する図である。
【図４５】zzzzz.clipのシンタクスを示す図である。
【図４６】ClipInfoのシンタクスを示す図である。
【図４７】Clip_stream_typeのテーブルを示す図である。
【図４８】offset_SPNについて説明する図である。
【図４９】offset_SPNについて説明する図である。
【図５０】ＳＴＣ区間について説明する図である。
【図５１】STC_Infoについて説明する図である。
【図５２】STC_Infoのシンタクスを示す図である。
【図５３】ProgramInfoを説明する図である。
【図５４】ProgramInfoのシンタクスを示す図である。
【図５５】VideoCondingInfoのシンタクスを示す図である。
【図５６】Video_formatのテーブルを示す図である。
【図５７】frame_rateのテーブルを示す図である。
【図５８】display_aspect_ratioのテーブルを示す図である。
【図５９】AudioCondingInfoのシンタクスを示す図である。
【図６０】audio_codingのテーブルを示す図である。
【図６１】audio_component_typeのテーブルを示す図である。
【図６２】sampling_frequencyのテーブルを示す図である。
【図６３】CPIについて説明する図である。
【図６４】CPIについて説明する図である。
【図６５】CPIのシンタクスを示す図である。
【図６６】CPI_typeのテーブルを示す図である。
【図６７】ビデオEP_mapについて説明する図である。
【図６８】EP_mapについて説明する図である。
【図６９】EP_mapについて説明する図である。
【図７０】EP_mapのシンタクスを示す図である。
【図７１】EP_type valuesのテーブルを示す図である。
【図７２】EP_map_for_one_stream_PIDのシンタクスを示す図である。

【図７３】TU_mapについて説明する図である。
【図７４】TU_mapのシンタクスを示す図である。
【図７５】ClipMarkのシンタクスを示す図である。
【図７６】mark_typeのテーブルを示す図である。
【図７７】mark_type_stampのテーブルを示す図である。
【図７８】menu.thmbとmark.thmbのシンタクスを示す図である。
【図７９】Thumbnailのシンタクスを示す図である。
【図８０】thumbnail_picture_formatのテーブルを示す図である。
【図８１】tn_blockについて説明する図である。
【図８２】DVR MPEG２のトランスポートストリームの構造について説明する図である。
【図８３】DVR MPEG２のトランスポートストリームのレコーダモデルを示す図である。
【図８４】DVR MPEG２のトランスポートストリームのプレーヤモデルを示す図である。
【図８５】source packetのシンタクスを示す図である。
【図８６】TP_extra_headerのシンタクスを示す図である。
【図８７】copy permission indicatorのテーブルを示す図である。
【図８８】シームレス接続について説明する図である。
【図８９】シームレス接続について説明する図である。
【図９０】シームレス接続について説明する図である
【図９１】シームレス接続について説明する図である。
【図９２】シームレス接続について説明する図である
【図９３】オーディオのオーバーラップについて説明する図である。
【図９４】BridgeSequenceを用いたシームレス接続について説明する図である。
【図９５】BridgeSequenceを用いないシームレス接続について説明する図である。
【図９６】DVR STDモデルを示す図である。　＊
＊【図９７】復号、表示のタイミングチャートを示す図である。
【図９８】BridgeSequenceInfoの他のシンタクスを示す図である。
【図９９】２つのPlayItemがシームレスに接続されるときのBridge-Clipについて説明する図である。
【図１００】ClipInformationファイルのシンタクスを示す図である。
【図１０１】ClipInformationファイルのClipInfoのシンタクスを示す図である。
【図１０２】ClipInformationファイルのSequenceInfoのシンタクスを示す図である。
【図１０３】ClipAVストリームファイルのストリームデータを部分的に消去した場合のデータベースの変更について説明する図である。
【図１０４】RealPlayListの作成について説明するフローチャートである。
【図１０５】VirtualPlayListの作成について説明するフローチャートである。
【図１０６】ブリッジシーケンスの作成について説明するフローチャートである。
【図１０７】PlayListの再生について説明するフローチャートである。
【図１０８】媒体を説明する図である。
【符号の説明】
１　記録再生装置，　１１乃至１３　端子，　１４　解析部，　１５　AVエンコーダ，　１６　マルチプレクサ，　１７　スイッチ，　１８　多重化ストリーム解析部，　１９　ソースパケッタイザ，　２０　ECC符号化部，　２１　変調部，　２２　書き込み部，　２３　制御部，　２４　ユーザインタフェース，２５　スイッチ，　２６　デマルチプレクサ，　２７　AVデコーダ，　２８　読み出し部，　２９　復調部，　３０　ECC復号部，　３１　ソースパケッタイザ，　３２，３３　端子

【図３】

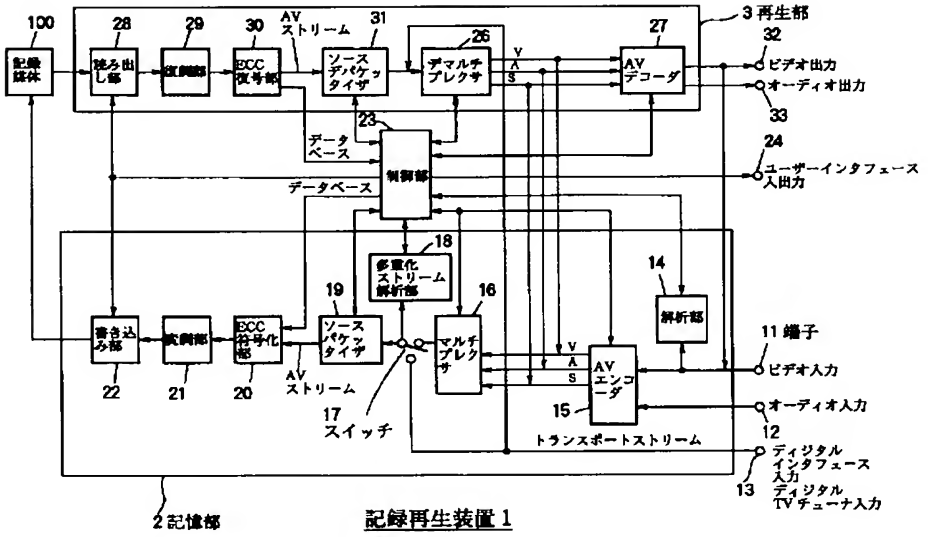


【図７】

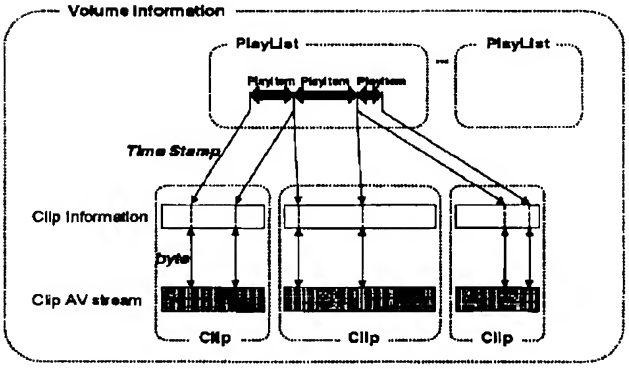


Virtual PlayList へのオーディオのアフレコの例

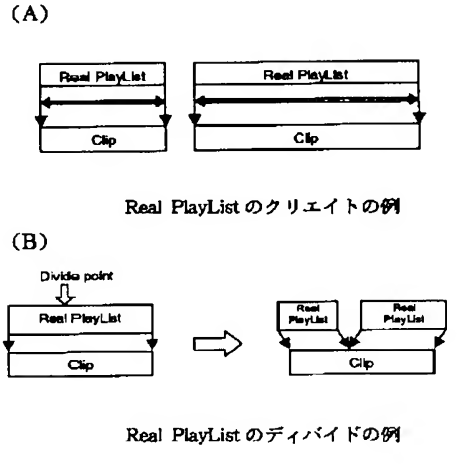
【図１】



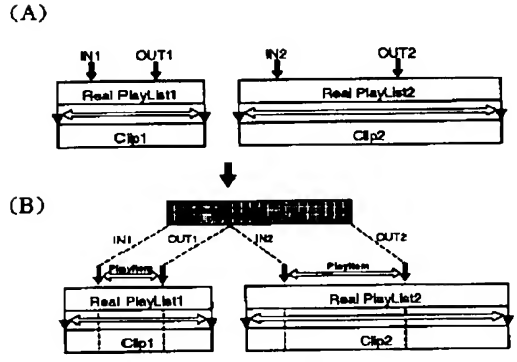
【図２】



【図４】



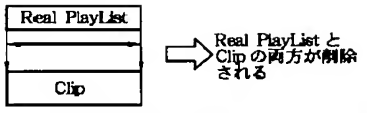
【図６】



アセンブル編集の例

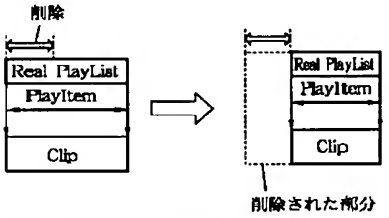
【図５】

(A)



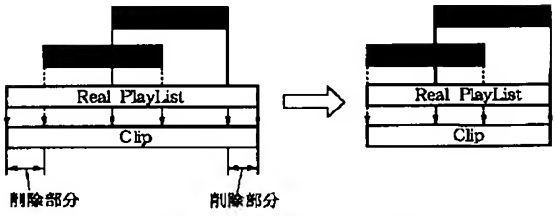
Real PlayList 全体のデリートの例

(B)



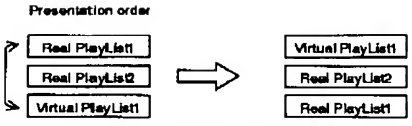
Real PlayList の部分的なデリートの例

(C)



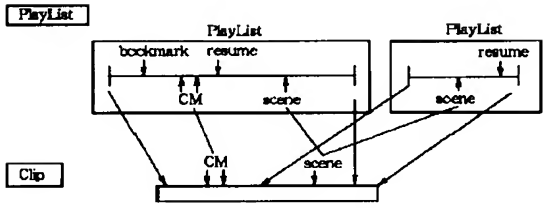
Real PlayList のミニマイズの例

【図８】



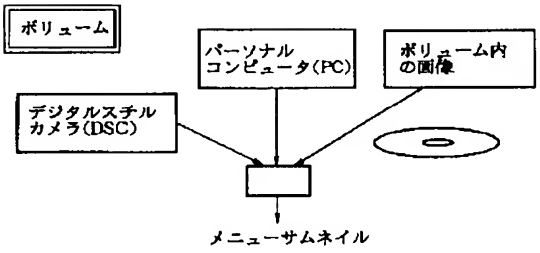
PlayList の再生順序の変更の例

【図９】

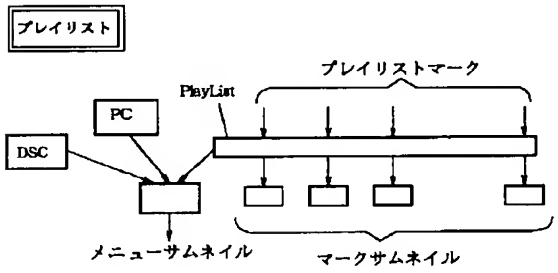


Playlist 上のマークと Clip 上のマーク

【図１０】



【図１１】

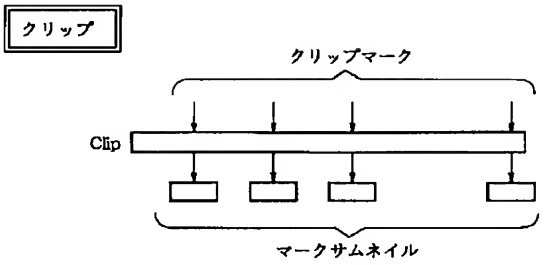


【図１９】

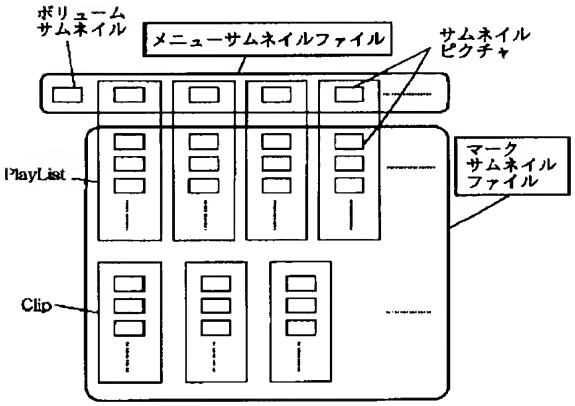
Value	Character coding
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

Character set value

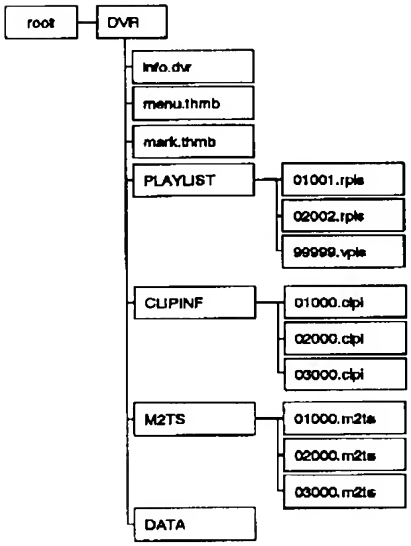
【図１２】



【図１３】



【図１４】



【図１５】

Syntax	No. of bits	Mnemonics
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakerPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

info.drv のシンタクス

【図１６】

Syntax	No. of bits	Mnemonics
DVRVolume() {		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

DVR Volume のシンタクス

【図２６】

PlayList_type	Meaning
0	AV記録のための PlayList この PlayList に参照されるすべての Clip は、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のための PlayList この PlayList に参照されるすべての Clip は、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2 - 255	reserved

PlayList_type

【図１７】

Syntax	No. of bits	Mnemonics
ResumeVolume() {		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

ResumeVolume のシンタクス

【図１８】

Syntax	No. of bits	Mnemonics
UIAppInfoVolume () {		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*255	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoVolume のシンタクス

【図２０】

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<*number_of_PlayLists*; i++) {		
PlayList_file_name	8*10	bslbf
}		
}		

TableOfPlayLists のシンタクス

【図２１】

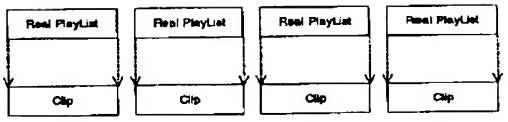
■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<*number_of_PlayLists*; i++) {		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

TableOfPlayLists の別シンタクス

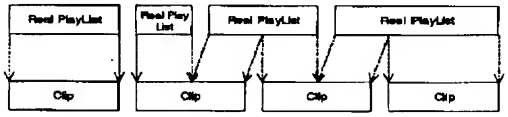
【図２４】

(A)



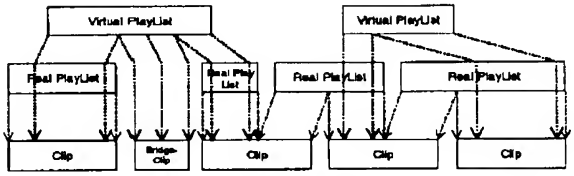
初めて AV ストリームが Clip として記録された時の Real PlayList の例

(B)



編集後の Real PlayList の例

(C)



Virtual PlayList の例

【図２８】

(A)

write_protect_flag	Meaning
0b	その PlayList を自由に消去しても良い。
1b	write_protect_flag を除いてその PlayList の内容は、消去および変更されるべきではない。

write_protect_flag

(B)

is_played_flag	Meaning
0b	その PlayList は、記録されてから一度も再生されたことがない。
1b	PlayList は、記録されてから一度は再生された。

is_played_flag

(C)

archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

archive

【図２２】

Syntax	No. of bits	Mnemonics
MakersPrivateData() {		
version_number	8*4	bslbf
length	32	uimsbf
if(length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<*number_of_maker_entries*; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (i=0; i<*number_of_mpd_blocks*; i++) {		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

MakersPrivateData のシンタクス

【図２３】

Syntax	No. of bits	Mnemonics
xxxxx.rpls / yyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakerPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

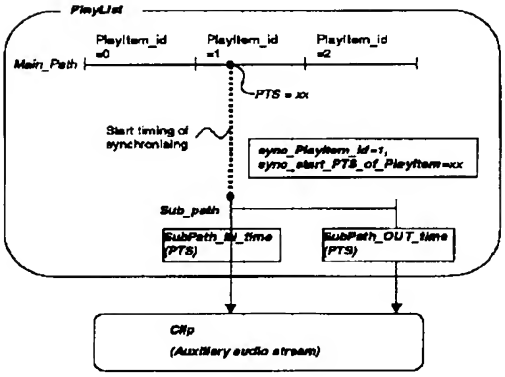
xxxxx.rpls と yyyyy.vpls のシンタクス

【図３２】

Syntax	No. of bits	Mnemonics
PlayItem() {		
Clip_Information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_Id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>) {		
if (*connection_condition*=='10') {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItem のシンタクス

【図３９】



【図４１】

SubPath_type	Meaning
0x00	Auxiliary audio stream path
0x01 - 0xff	reserved

SubPath_type

【図２５】

Syntax	No. of bits	Mnemonics
PlayList() {		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<*Virtual PlayList*>) {		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (*PlayItem_id=0; PlayItem_id<number_of_PlayItems; PlayItem_id++*) {		
PlayItem() // main path		
}		
if (<*Virtual PlayList*>) {		
if (CPI_type==0 && PlayList_type==0) {		
for (i = 0; i < *number_of_SubPlayItems*; i++)		
SubPlayItem() // sub path		
}		
}		
}		

PlayList のシンタクス

【図２７】

Syntax	No. of bits	Mnemonics
UIAppInfoPlayList2 ( ) {		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*8	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoPlayList のシンタクス

【図３３】

CPI_type in the PlayList()	Semantics of IN_time
EP_map type	IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	IN_time は、*TU_map_time_axis* 上の時刻でなければならない。かつ、IN_time は、time_unit の精度に丸めて表さねばならない。IN_time は、次に示す等式により計算される。 $IN\_time = TU\_start\_time \% 2^{32}$

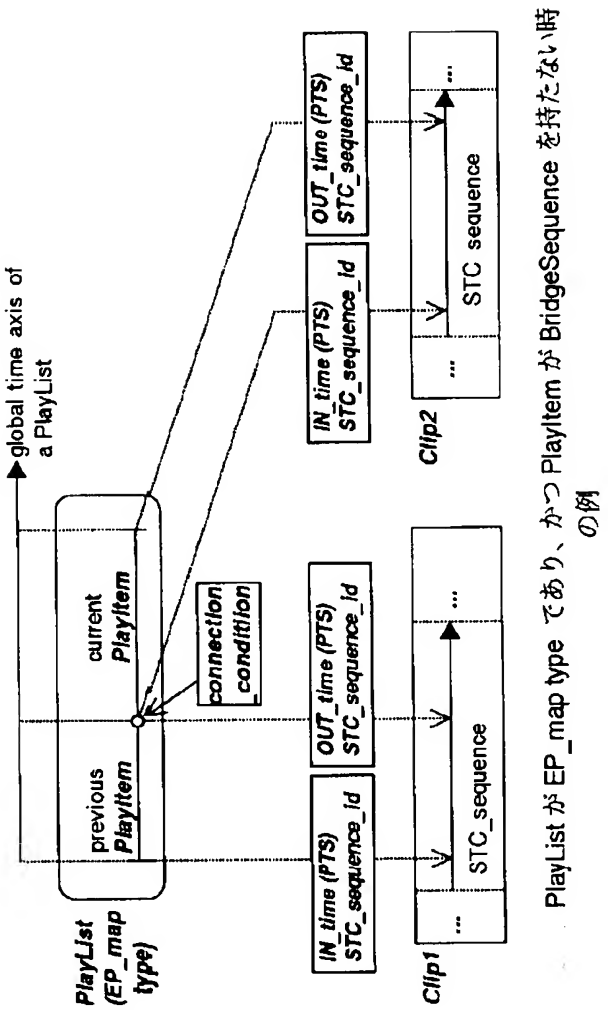
IN_time

【図４７】

Clip_stream_type	meaning
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2 - 255	Reserved

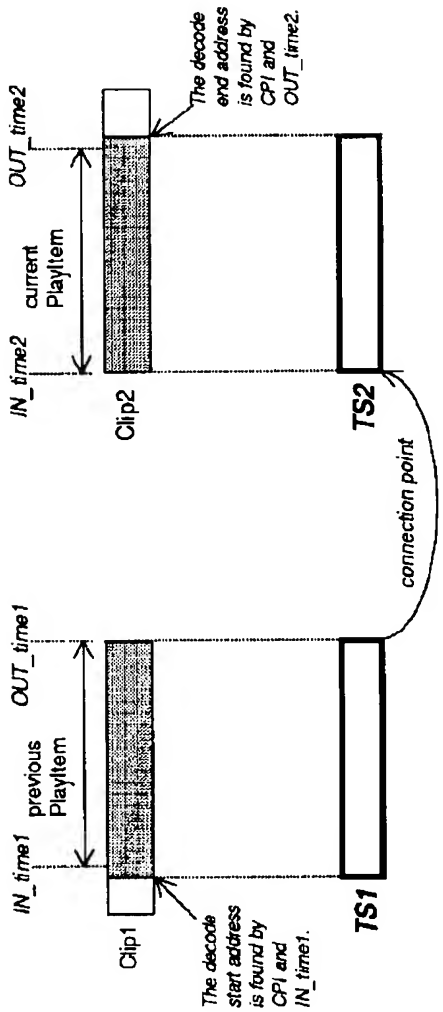
Clip_stream_type

【図２９】

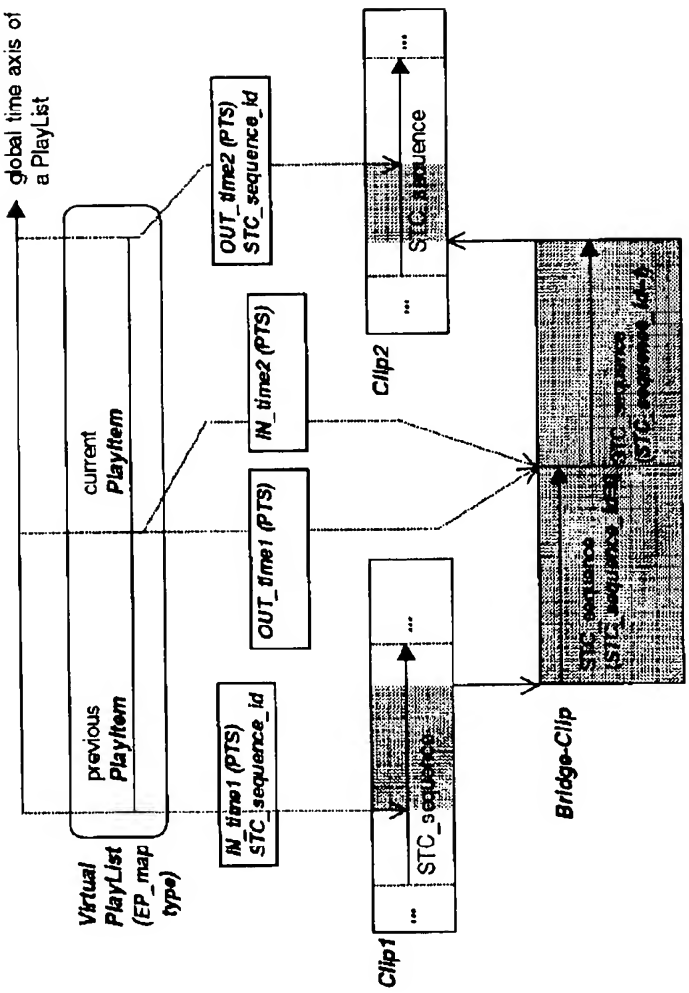


PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持たない時の例

【図８９】

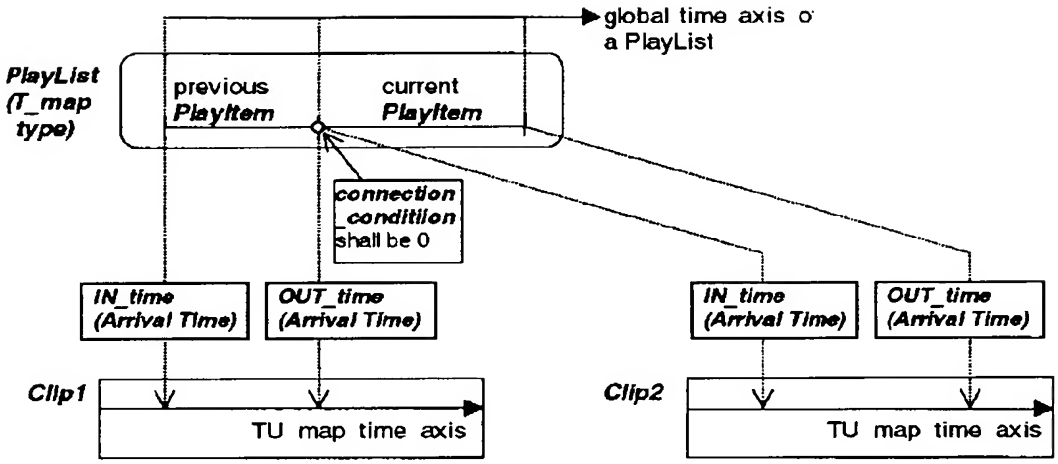


【図３０】



PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持つ時の例

【図３１】



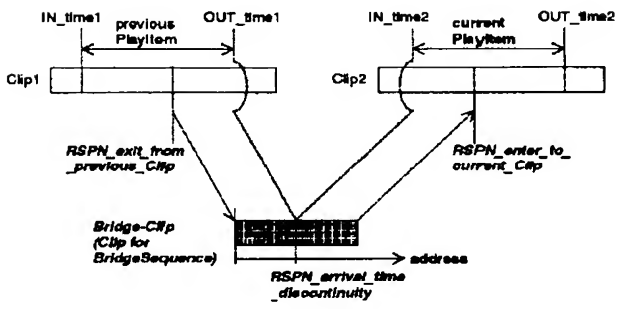
PlayList が TU_map type である時の例

【図３４】

CPI_type in the PlayList()	Semantics of OUT_time
EP_map type	OUT_time は、次に示す等式によって計算される Presenation_end_TS の値の上位 32 ビットを示さなければならない。 *Presentation_end_TS = PTS_out + AU_duration* ここで、 *PTS_out* は、PlayItem の中で最後のプレゼンテーションユニットに対応する 33 ビット長の PTS である。 *AU_duration* は、最後のプレゼンテーションユニットの 90kHz 単位の表示期間である。
TU_map type	OUT_time は、*TU_map_time_axis* 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さねばならない。OUT_time は、次に示す等式により計算される。 $OUT\_time = TU\_start\_time \% 2^{32}$

OUT_time

【図３７】



【図６６】

CPI_type	Meaning
0	EP_map type
1	TU_map type

CPI_type の意味

【図４４】

CPI_type in the PlayList()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、*TU_map_time_axis* 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さねばならない。mark_time_stamp は、次に示す等式により計算される。 $mark\_time\_stamp = TU\_start\_time \% 2^{32}$

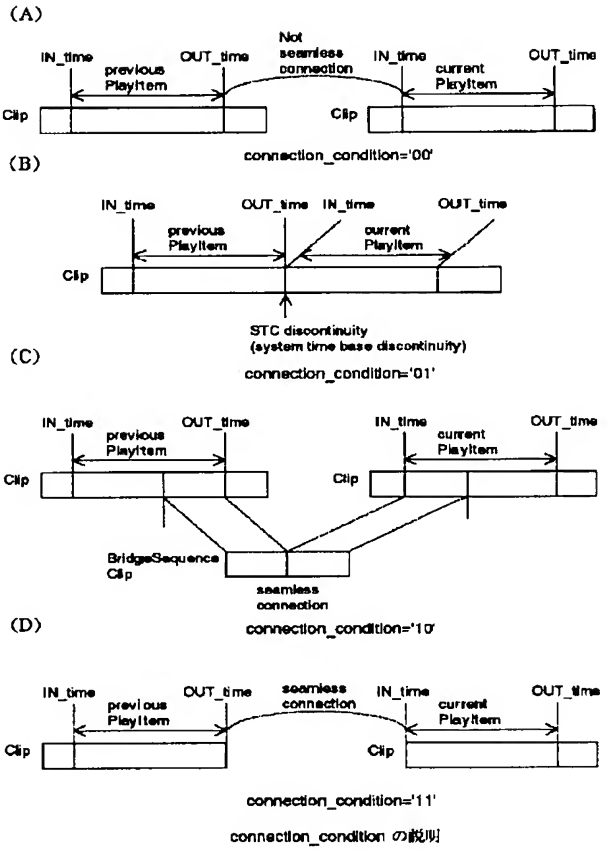
mark_time_stamp

【図３５】

connection_condition	meaning
00	• 先行する PlayItem と現在の PlayItem の接続は、シームレス再生の保証がなされていない。 • PlayList の CPI_type が TU_map type である場合、connection_condition は、この値をセットされねばならない。
01	• この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 • 先行する PlayItem と現在の PlayItem は、システムタイムベース（STC ベース）の不連続点があるために分割されていることを表す。
10	• この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 • この状態は、Virtual PlayList に対してだけ許される。 • 先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされている。 • 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用して接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていなければならない。
11	• この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 • 先行する PlayItem と現在の PlayItem は、シームレス再生の保証がなされている。 • 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用しないで接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていなければならない。

connection_condition

【図３６】



connection_condition の説明

【図３８】

Syntax	No. of bits	Mnemonics
BridgeSequenceInfo() {		
Bridge_Clip_Information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

BridgeSequenceInfo のシンタクス

【図４０】

Syntax	No. of bits	Mnemonics
SubPlayItem() {		
Clip_information_file_name	6*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

SubPlayItem のシンタクス

【図４２】

Syntax	No. of bits	Mnemonics
PlayListMark() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for(i=0; i < *number_of_PlayList_marks*; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

PlayListMark のシンタクス

【図４３】

Mark_type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、０または１でなければならない。
0x01	book-mark	PlayList の再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark() において定義されるスキップマークポイントの数は、０または１でなければならない。
0x03 - 0x8F	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark_type

【図６２】

sampling_frequency	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling_frequency

【図５６】

video_format	Meaning
0	480i
1	576i
2	480p (including 640x480p format)
3	1080i
4	720p
5	1080p
6 - 254	reserved
255	No information

vidoe_format

【図４５】

Syntax	No. of bits	Mnemonics
zzzzz.clpi {		
STC_info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakerPrivateData_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
STC_Info()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0; i<N3; i++){		
padding_word	16	bslbf
}		
CPI()		
for(i=0; i<N4; i++){		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0; i<N5; i++){		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

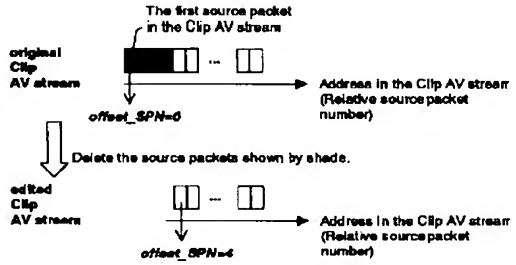
zzzzz.clpi のシンタクス

【図４６】

Syntax	No. of bits	Mnemonics
ClipInfo() {		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_future_use	256	bslbf
}		

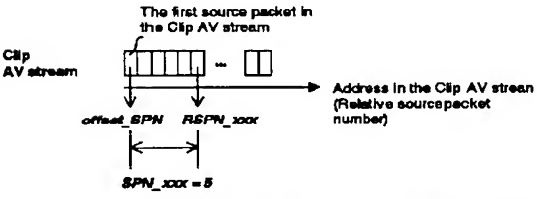
ClipInfo のシンタクス

【図４８】



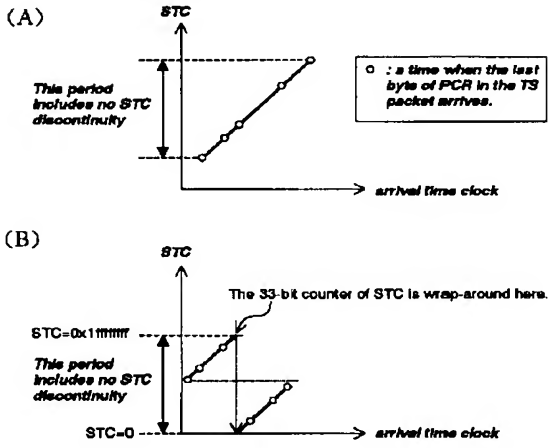
offset_SPN がゼロ以外の値をとる場合の例

【図４９】



AV ストリームでの offset_SPN と相対ソースパケット番号（RSPN_xxx）の間の関係

【図５０】

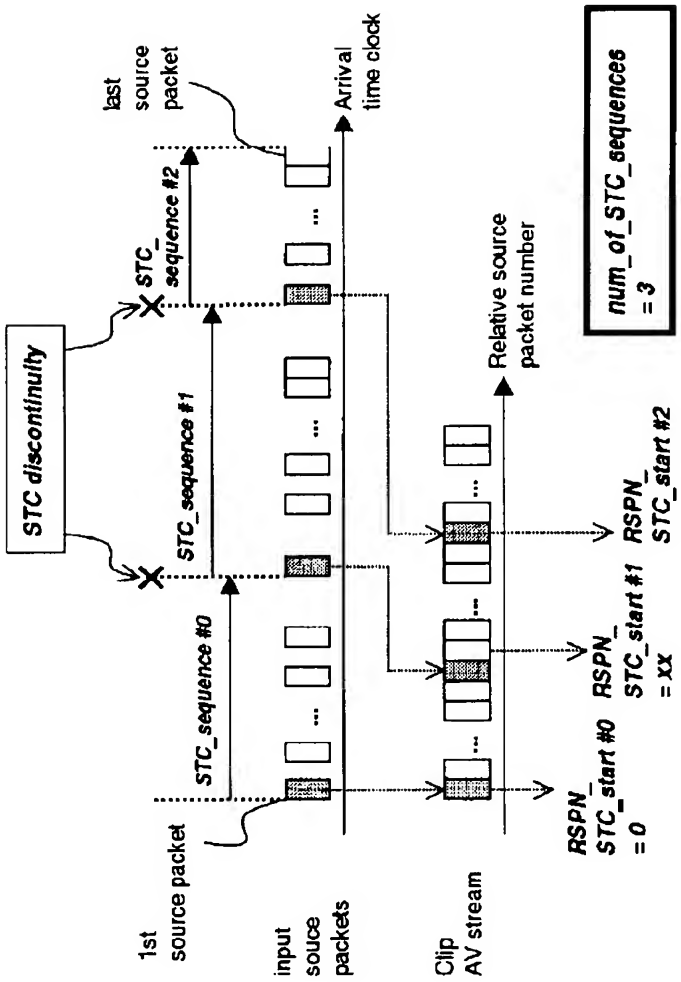


【図６１】

audio_component_type	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

audio_component_type

【図５１】


last source packet
Arrival time clock
STC_ sequence #2
STC discontinuity
STC_sequence #1
STC_sequence #0
Relative source packet number
num_of_STC_sequences = 3
RSPN_ STC_start #2 = yy
RSPN_ STC_start #1 = xx
RSPN_ STC_start #0 = 0
1st source packet
input souce packets
Clip AV stream

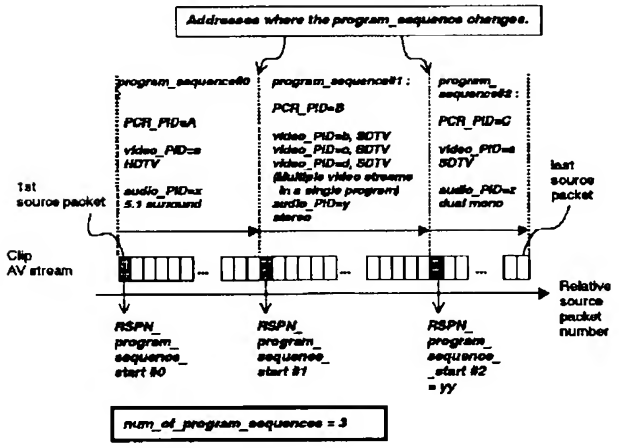
--STC_Info

【図５２】

Syntax	No. of bits	Mnemonics
STC_Info() {		
version_number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for(*STC_sequence_id=0;* *STC_sequence_id < num_of_STC_sequences;* *STC_sequence_id++)* {		
reserved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

STC_Info のシンタクス

【図５３】



ProgramInfo の例

【図５５】

Syntax	No. of bits	Mnemonics
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

VideoCodingInfo のシンタクス

【図５７】

frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97...)
5	30
6	50
7	60 000/1001 (59.94 ...)
8	60
9 - 254	reserved
255	No information

frame_rate

【図５８】

display_aspect_ratio	Meaning
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

display_aspect_ratio

【図５４】

Syntax	No. of bits	Mnemonics
ProgramInfo() {		
version_number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for(i=0; i<*number_of_program_sequences*; i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0; k<*number_of_videos*; k++) {		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0; k<*number_of_audios*; k++) {		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

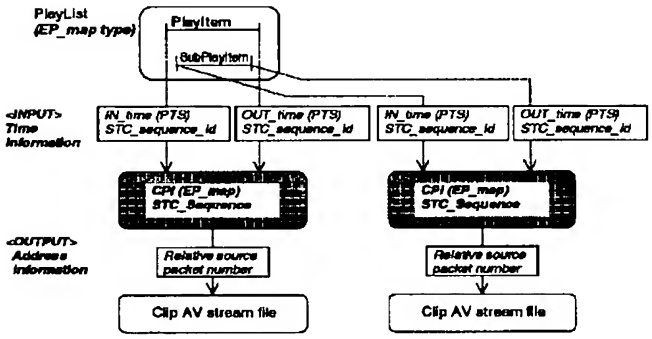
ProgramInfo のシンタクス

【図５９】

Syntax	No. of bits	Mnemonics
AudioCodingInfo() {		
audio_coding	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

AudioCodingInfo のシンタクス

【図６３】



【図８０】

Thumbnail_picture_format	*Meaning*
0x00	*MPEG-2 Video I-picture*
0x01	*DCF (restricted JPEG)*
0x02	*PNG*
0x03-0xff	*reserved*

thumbnail__picture__format

【図６０】

audio_coding	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

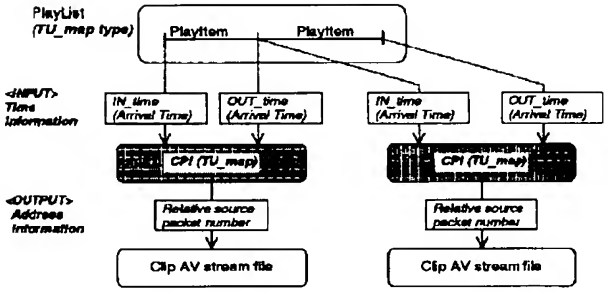
audio_coding

【図７１】

EP_type	Meaning
0	video
1	audio
2 - 15	reserved

EP_type Values

【図６４】

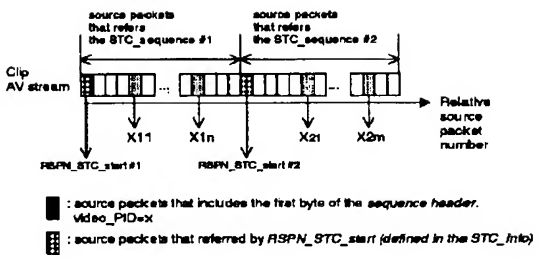


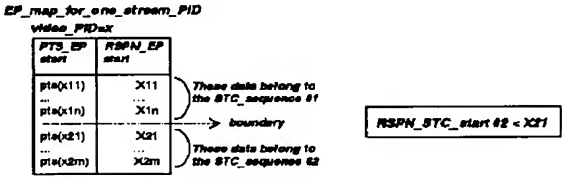
【図６５】

Syntax	No. of bits	Mnemonics
CPI() {		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type == 0)		
EP_map()		
else		
TU_map()		
}		

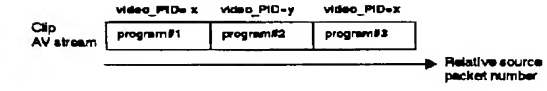
CPI のシンタクス

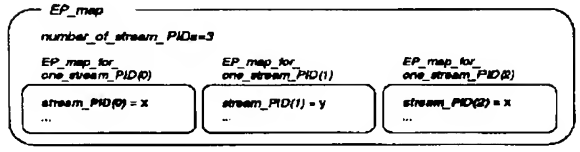
【図６８】





【図６９】



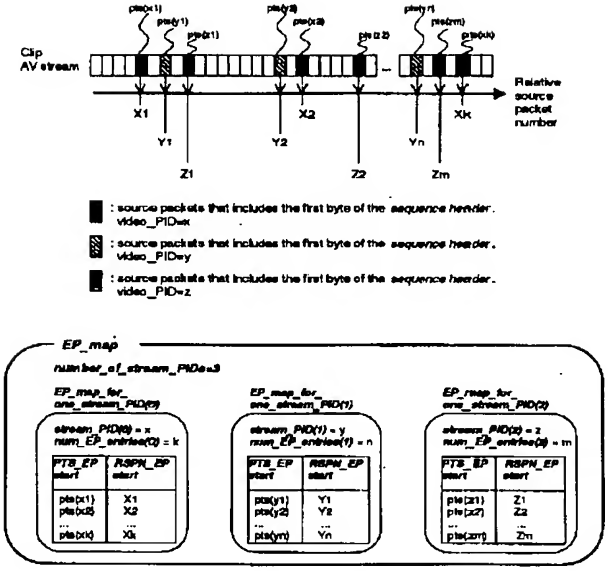


【図７２】

Syntax	No. of bits	Mnemonics
EP_map_for_one_stream_PID(N){		
for (i=0; i< N; i++) {		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

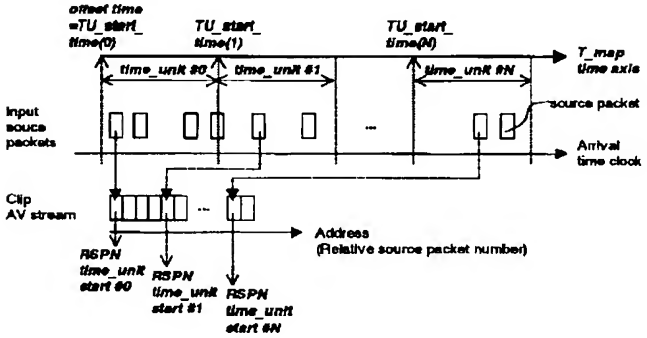
EP_map_for_one_stream_PID のシンタクス

【図６７】



ビデオの EP_map の例

【図７３】



【図７４】

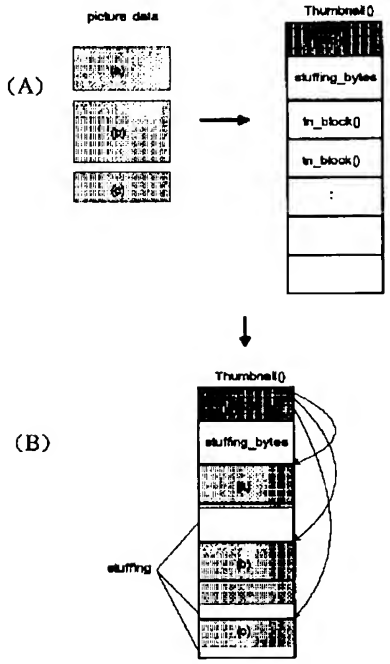
Syntax	No. of bits	Mnemonics
TU_map(){		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0; k<number_of_time_unit_entries ; k++)		
RSPN_time_unit_start	32	uimsbf
}		

TU_map のシンタクス

【図７０】

Syntax	No. of bits	Mnemonics
EP_map(){		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID (k)	16	bslbf
num_EP_entries (k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address (k)	32	uimsbf
}		
for(i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for(i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

【図８１】



【図８７】

copy_permission indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

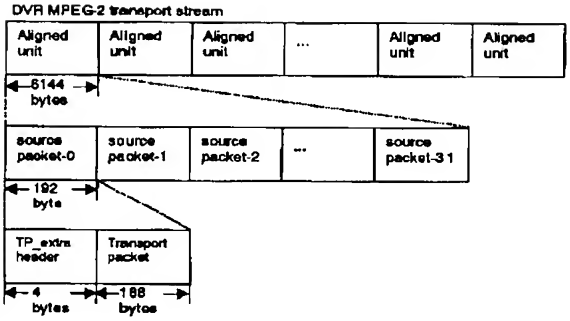
· copy permission indicator table

【図７５】

Syntax	No. of bits	Mnemonics
ClipMark() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for(i=0; i < *number_of_Clip_marks*; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

ClipMark のシンタクス

【図８２】



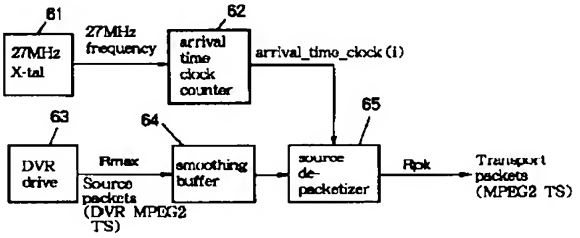
DVR MPEG-2　トランスポートストリームの構造

【図７６】

Mark_type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点。
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x92	Scene-start mark	シーンチェンジポイントを示すマーク。
0x93 - 0xFF	reserved	

mark_type

【図８４】



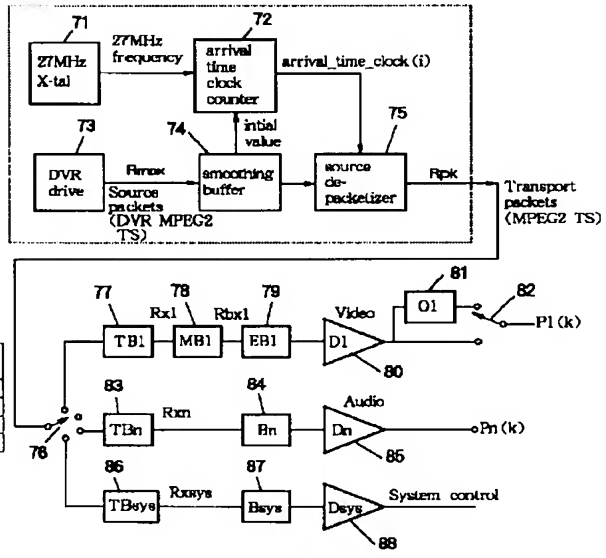
DVR MPEG-2　トランスポートストリームのプレーヤモデル

【図７７】

CPI_type in the CPI()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、*TU_map_time_axis* 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さねばならない。mark_time_stamp は、次に示す等式により計算される。 mark_time_stamp = *TU_start_time* % 2^{32}

mark_type_stamp

【図９６】



【図８５】

Syntax	No. of bits	Mnemonics
source_packet() {		
TP_extra_header()		
transport_packet()		
}		

source packet

【図７８】

Syntax	No. of bits	Mnemonics
menu.thmb / mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for(i=0; i<N1; i++)		
padding_word	16	bslbf
}		

menu.thmb と mark.thmb のシンタクス

【図７９】

Syntax	Bits	Mnemonics
Thumbnail() {		
version_number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for(i = 0; i < *number_of_thumbnails*; i++) {		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k = 0; k <*number_of_tn_blocks*; k++) {		
tn_block	tn_block_size* 1024*8	
}		
}		
}		

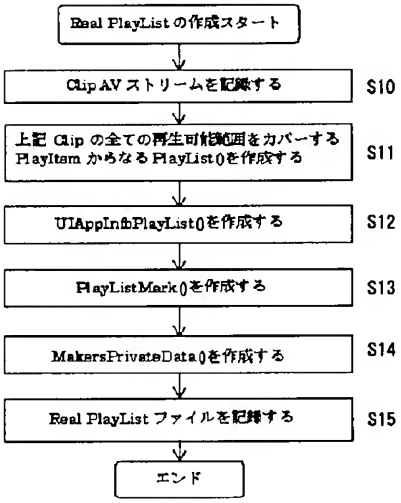
Thumbnail のシンタクス

【図８６】

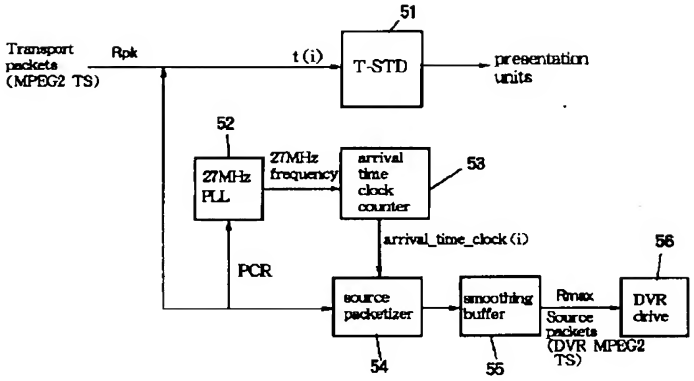
Syntax	No. of bits	Mnemonics
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

TP_extra_header

【図１０４】

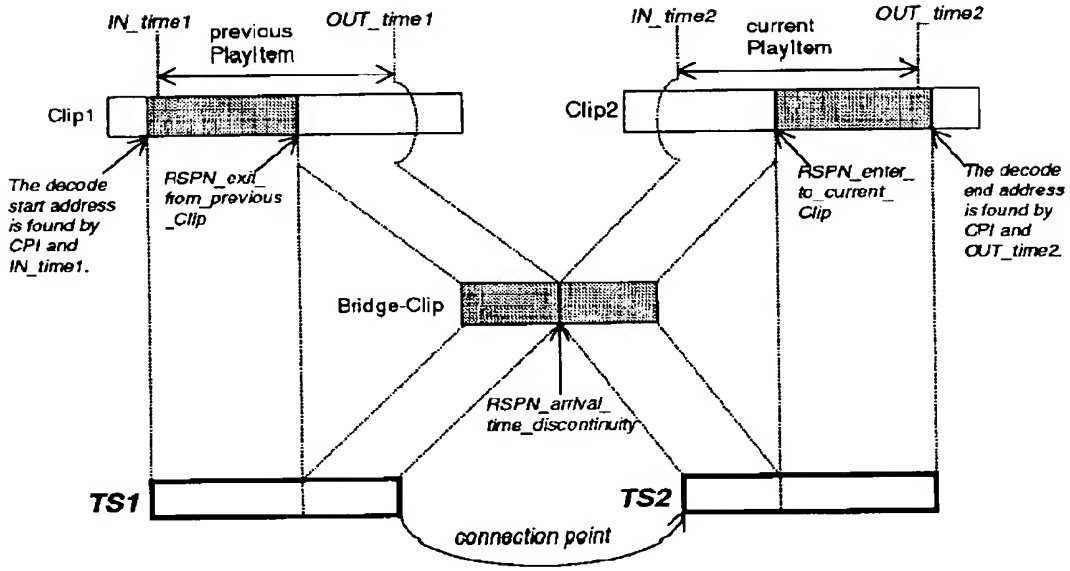


【図８３】



DVR MPEG-2　トランスポートストリームのレコーダモデル

【図８８】

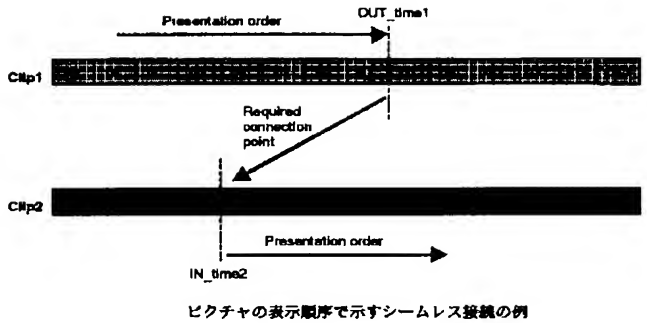


【図９８】

Syntax	No. of bits	Mnemonic
BridgeSequenceInfo(){		
Bridge_Clip_Information_file_name	8*10	bslbf
}		

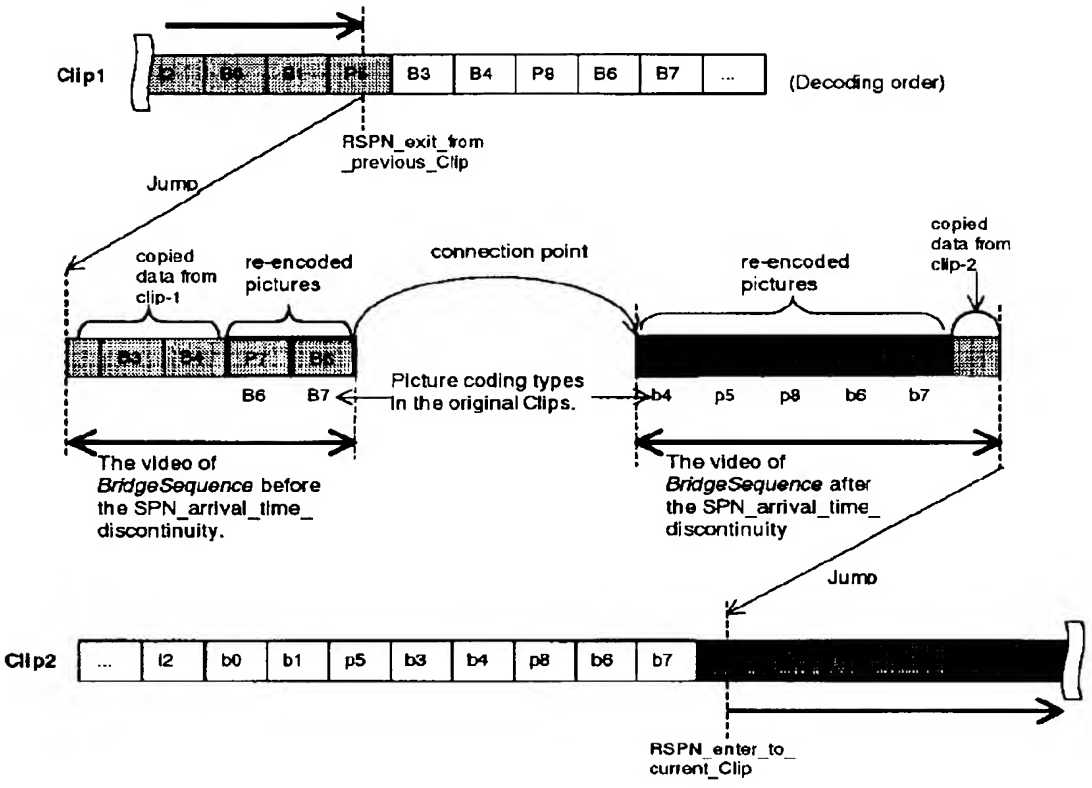
BridgeSequenceInfo()のシンタクス

【図９０】



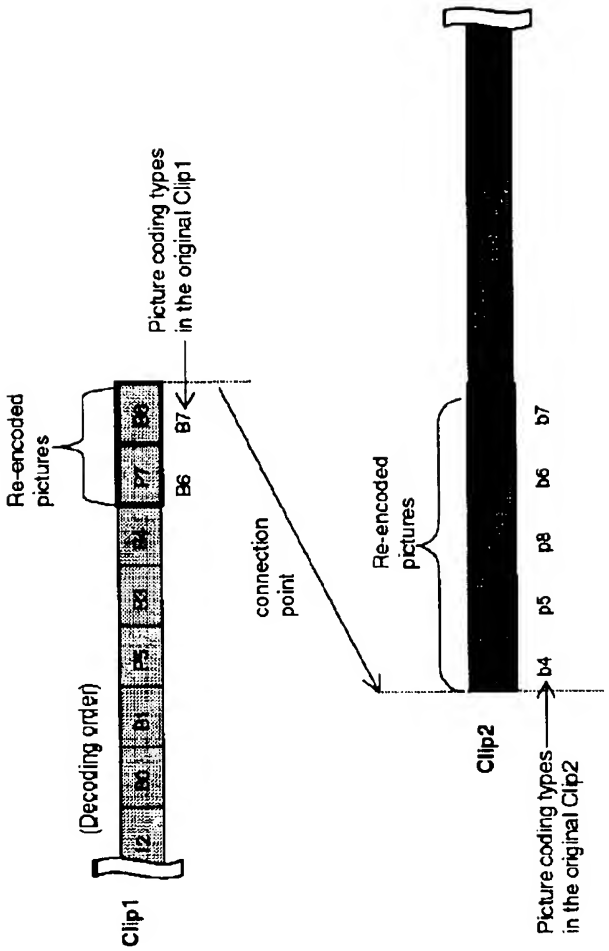
ピクチャの表示順序で示すシームレス接続の例

【図９１】



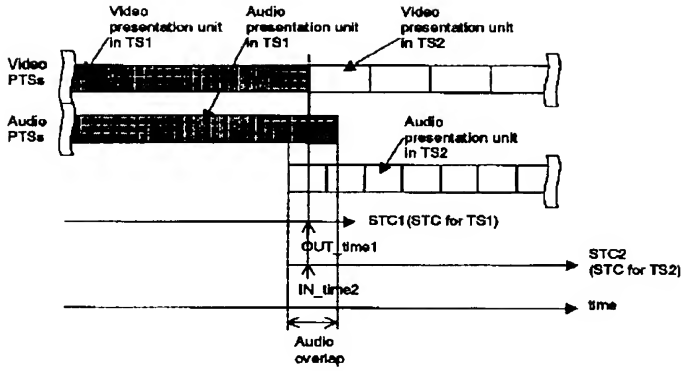
BridgeSequence を使用してシームレス接続を実現する例１

【図９２】

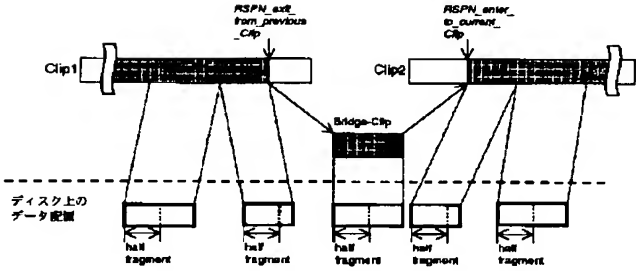


BridgeSequence を使用しないでシームレス接続を実現する例２

【図９３】

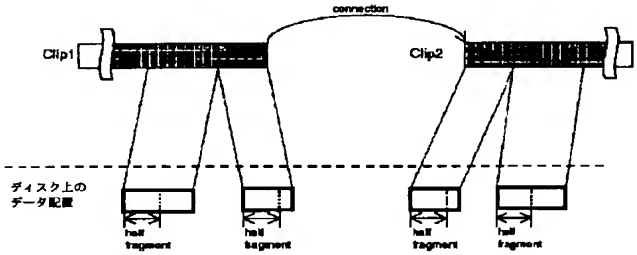


【図９４】



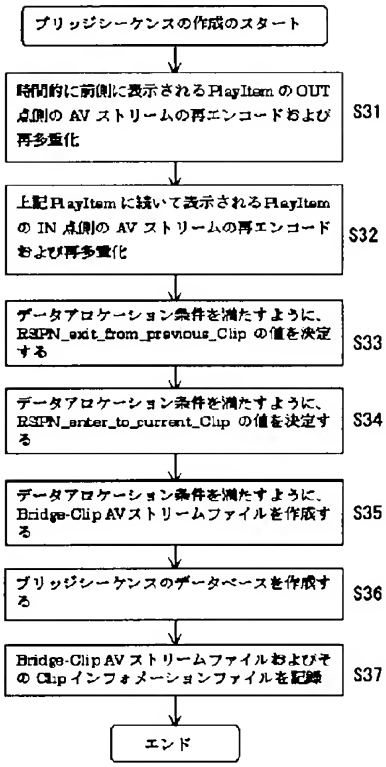
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図９５】

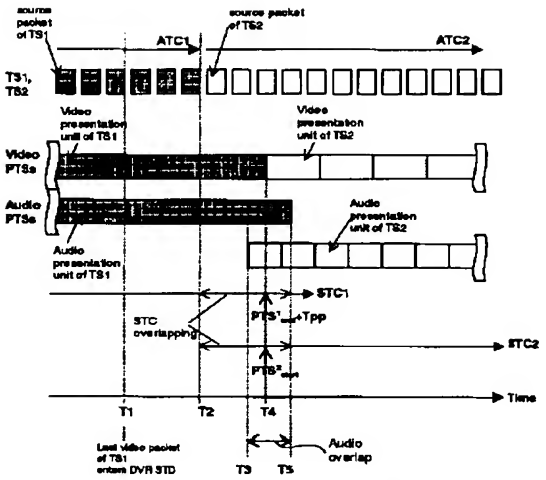


BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

【図１０６】

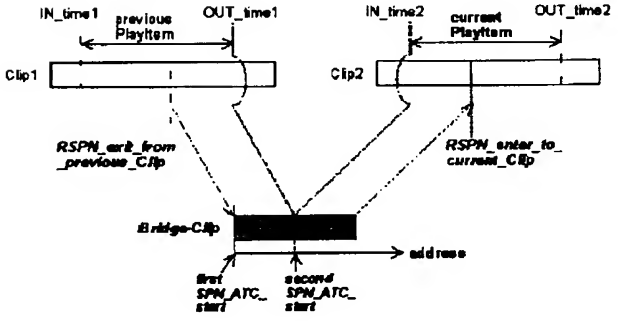


【図９７】



あるAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

【図９９】



【図１００】

Syntax	No. of bits	Mnemonic
zzzzz.clpi{		
version_number	8*4	bslbf
SequenceInfo_start_address	32	uimsbf
ProgramInfo_start_address	32	uimsbf
CPI_start_address	32	uimsbf
ClipMark_start_address	32	uimsbf
MakersPrivateData_start_address	32	uimsbf
reserved_for_future_use	96	bslbf
ClipInfo()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
SequenceInfo()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0; i<N3; i++){		
padding_word	16	bslbf
}		
CPI()		
for(i=0; i<N4; i++){		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0; i<N5; i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for(i=0; i<N6; i++){		
padding_word	16	bslbf
}		
}		

Clip Informationファイルのシンタクス

【図１０１】

Syntax	No. of bits	Mnemonic
ClipInfo() {		
length	32	uimsbf
reserved_for_future_use	16	bslbf
Clip_stream_type	8	uimsbf
reserved_for_word_align	6	bslbf
transcode_mode_flag	1	bslbf
controlled_time_flag	1	bslbf
TS_average_rate	32	uimsbf
TS_recording_rate	32	uimsbf
reserved_for_DVRsystem_use	144	bslbf
TS_type_info_block()		
if (Clip_stream_type == "Bridge-Clip AV stream") {		
previous_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
current_Clip_information_file_name	8*10	bslbf
RSPN_enter_to_current_Clip	32	uimsbf
}		
}		

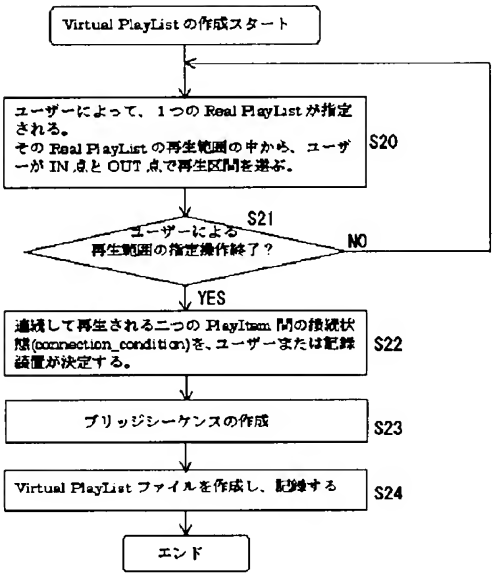
Clip InformationファイルのClipInfo()のシンタクス

【図１０２】

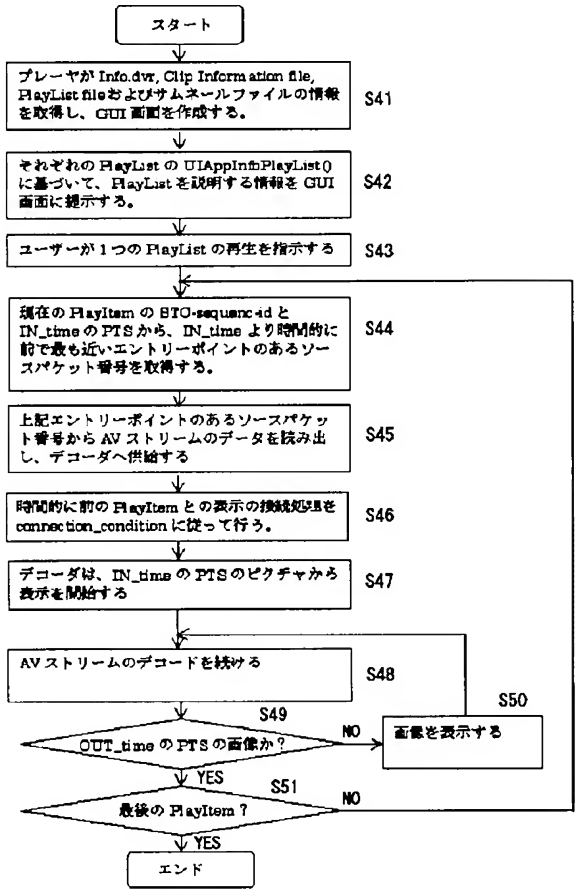
Syntax	No. of bits	Mnemonic
SequenceInfo() {		
length	32	uimsbf
reserved_for_word_align	8	bslbf
num_of_ATC_sequences	8	uimsbf
for (atc_id=0; atc_id<num_of_ATC_sequences; atc_id++) {		
SPN_ATC_start[atc_id]	32	uimsbf
num_of_STC_sequences[atc_id]	8	uimsbf
offset_STC_id[atc_id]	8	uimsbf
for (stc_id = offset_STC_id[atc_id]; stc_id <(num_of_STC_sequences[atc_id]+offset_STC_id[atc_id]); stc_id++) {		
PCR_PID[atc_id][stc_id]	16	uimsbf
SPN_STC_start[atc_id][stc_id]	32	uimsbf
presentation_start_time[atc_id][stc_id]	32	uimsbf
presentation_end_time[atc_id][stc_id]	32	uimsbf
}		
}		

Clip InformationファイルのSequenceInfo()シンタクス

【図１０５】

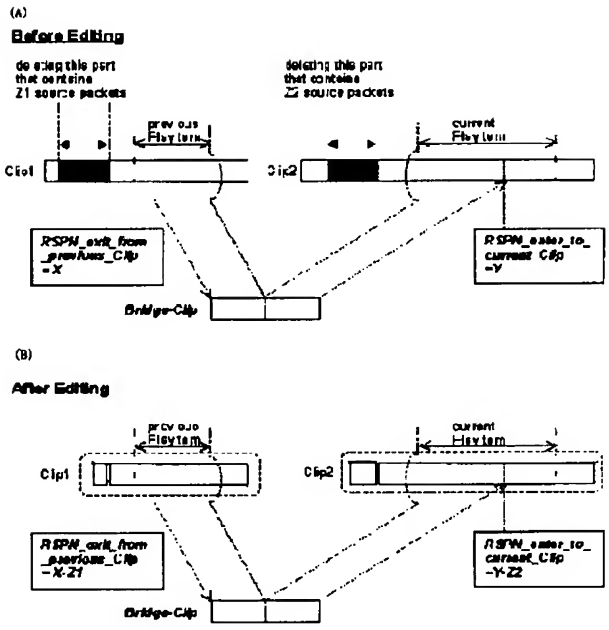


【図１０７】

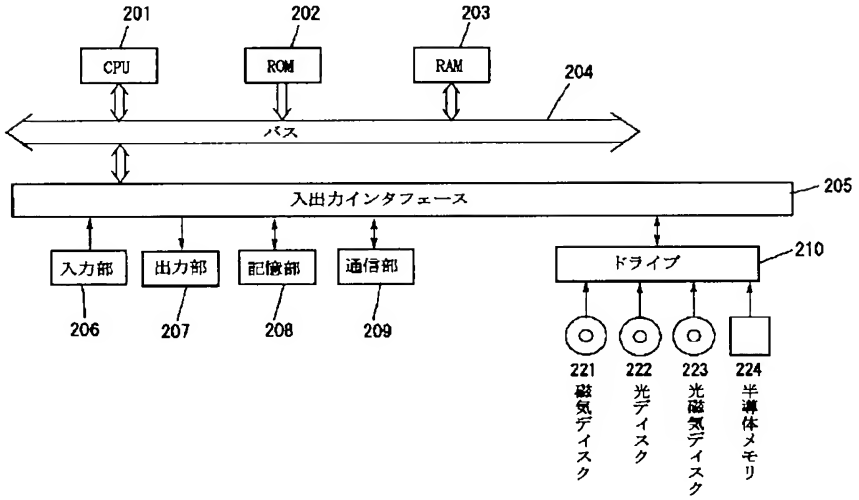


【図１０３】

図103



【図１０８】



フロントページの続き

(51)Int.Cl.[7]	識別記号	F I		テーマコード（参考）
H04N 5/92		H04N 5/92		H
7/24		7/13		Z

Ｆターム(参考) 5C052 AA02 AC01 CC11 FA05
5C053 FA24 GA11 GB04 GB17 GB21
GB37 HA21 JA24
5C059 KK32 MA00 RB02 RB09 RC04
RF05 SS11 SS20 UA05 UA36
5D044 AB07 BC03 CC06 DE25 DE28
DE38 DE96 EF05 FG18 FG23
GK08 GK12